Rank Based Moth Flame Optimisation for Feature Selection in the Medical Application

Ruba Abu Khurma, Ibrahim Aljarah, Ahmad Sharieh
King Abdullah II School for Information Technology
The University of Jordan
Amman, Jordan
RBA9150539@FGS.ju.edu.jo{i.aljarah,sharieh}@ju.edu.jo

***Abstract*—Feature selection (FS) is a challenging data mining problem that incorporates a complex search process to find the most informative feature subset. In the brute force methods generating the entire feature space and applying an exhaustive search makes the FS NP-hard problem. Meta-heuristic algorithms are good alternative solutions that provide (near) optimal solutions through a random search process instead of a complete search. In this paper, an FS approach based on the Moth Flame Optimization algorithm (MFO) and k-NN classifier are proposed. MFO is a recent meta-heuristic algorithm that has proved its effectiveness in solving different complex problems in a reasonable time. Nevertheless, the performance of MFO highly depends on achieving a balance between exploration and exploitation during the search process. To address this issue, we propose an adaptive method to update the position of a moth toward the best global solution based on the search status. The proposed MFO has been evaluated using sixteen benchmark medical data sets and the results show promising performance of the modified MFO algorithm in terms of the applied evaluation measures.**

***Index Terms*—Feature Selection, Classification, Moth Flame Optimisation.**

I. Introduction

Advances in data collection technologies have produced huge data sets with a massive number of dimensions (features). In the data mining community, this is known as the curse of dimensionality phenomenon. It causes several negative consequences for the learning process including slowing down the learning time and degrading the learner's performance [1]. Feature selection (FS) is a dimensionality reduction technique that produces a smaller version of a data set without affecting the original meaning of features. This is done by eliminating the noisy features and maintaining the most representative features that are highly correlated with the target class (relevant) and weakly correlated with each other (not redundant). The target of using FS as a preprocessing step in a data mining task (e.g. classification or clustering) is to achieve two conflicting objectives simultaneously: minimizing the number of features and maximizing the performance of the learning algorithm.

FS process traditionally consists of two main processes: search and evaluation [45]–[49]. In the evaluation process, each candidate feature subset is assessed to determine its suitability as a solution for an FS problem. Two main methods that can be used to evaluate a feature subset: filters and wrappers. Filters rely on the properties of the data set itself without involving any learning step so it is a simple and fast method such as F-score, Information Gain (IG) and Chi-square [42]. On the other hand, wrappers incorporate a learning step in the FS process that consumes a longer time, but it may contribute to better performance results.

Search in an FS process means moving around in the feature space to find the best feature subset among the generated feature subsets. This can be done by either creating the entire feature space using a complete search algorithm or by generating random feature subsets using a meta-heuristic search algorithm (MH). Applying the complete search and exhaustively traversing all the feature subsets generated from a moderate and large data set is impractical. In mathematics, if the size of the data set is N dimensions then the size of the fully generated feature space size is 2^N. This requires exponential running time which makes FS NP-hard problem. Applying the MH algorithm reduces the feature space complexity and efficiently guides the search procedure for a (near) optimal solution.

The main category of MH algorithms is Swarm Intelligence (SI). SI algorithm simulates the natural survival behaviors of creatures that live in groups. The method of exchanging information between group members to approach prey is transformed into mathematical models like Grey wolf optimization (GWO) [41], Binary Cuckoo Search (BCS) [33], and Binary Bat Algorithm (BBA) [32]

Over the past decade, there has been a monotonous increase in the use of SI algorithms to solve various optimization problems including the FS problem. A wide range of applications have gained the benefits of FS-SI approaches including: facial expression recognition [16], Arabic handwritten letter recognition [7], hyperspectral image processing [8], protein and related genome annotation [9], biochemistry and drug design [10], electroencephalogram (EEG) [4], financial diagnosis [5], [6], software product line estimation [11], spam detection in emails [12] and medical application [13]. In the medical applications, the FS-SI approach has been widely applied to improve the classification tasks by preprocessing the medical data set without affecting its readability and changing the original features. In literature, there were many studies that

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

proposed different modification strategies to enhance SIs to solve FS problem such as integrating new operators (e.g. chaotic maps [44], rough set [15], transfer functions [28] and Levy flight [16]), hybridization with other algorithm (e.g. filter [1], SI [17] and classifier [31]), using new initialization mechanism [18] and adopting new update strategy such as the time-varying strategy [28].

The MFO is a recent SI algorithm that is inspired by the behavior of moths in nature [19]. MFO as all SI algorithms follows a population-based paradigm that requires the initialization of a population of moths at the initial optimization steps. Each moth represents a possible solution for the optimization problem at hand. MFO applies a spiral update strategy to change the positions of moths in the solution space. Moths are updated and evaluated in an iterative process until a stopping condition is satisfied (e.g maximum number of iteration is reached). In MFO, the optimization process is performed in two phases: exploration and exploitation. In exploration, the moths distribute globally to explore different regions of the search space. In exploitation, the moths search locally within a limited region where the global optimal solution may exist. A smooth transition between these two conflicting phases during the optimization process improves the performance of the search algorithm. This is because if the exploration phase takes longer, this leads to the loss of the optimal solution and an increase in search time, while if the exploitation is longer, this will cause stagnation in the local minimum.

MFO was proposed in [19] to solve continuous optimization problems, then a binary version called BMFO was proposed in [35] to solve the binary optimization problem such as FS. Since then, many modifications have been adopted to improve the BMFO as a search algorithm in the FS process including [16], [29]–[31]. The reason for this widespread usage of MFO as a wrapper FS approach is that the MFO is equipped with spiral update strategy and adaptive parameters that control the adaptive convergence of MFO and maintain the trade-off between exploration and exploitation. For more information about MFO, a reader can refer to the reviews [36], [37]. By referring to No Free Lunch Theorem (NFL) [43], there is no evolutionary algorithm that is considered the perfect solution for all optimization problems. Therefore, the doors are always opened to either suggest new evolutionary algorithms or to suggest new modifications to improve existing algorithms when used to solve various optimization problems.

In this paper, a new adaptive update mechanism is proposed to update the position of a moth toward the flame based on the search status. Instead of the original update strategy that updates all moths in the same way and ignores their closeness to the global solution. In this work, each moth will be updated adaptively by taking into account its position from the global solution. Thus, all moths in the swarm will be given a rank that depends on how far a moth from the global solution. Therefore, a low-quality moth with a low fitness value will have a large rank and there will be a large change in its position. On the other hand, a high-quality moth with a high fitness value will have a small rank and there will be a small change in its position. To evaluate the performance of the proposed MFO, sixteen benchmark medical data sets were used and the results are compared with three similar approaches from the literature. The results showed promising performance of the proposed MFO in terms of the applied evaluation measures. The remaining of this paper is organized as follows: Section II presents the methodology of the MFO algorithm and its binary version. Section III discusses the proposed FS approach. In Section IV, the experimental results are discussed and analyzed. Finally, in Section V, conclusions are provided.

II. METHODOLOGY

A. Moth Flame Optimisation (MFO)

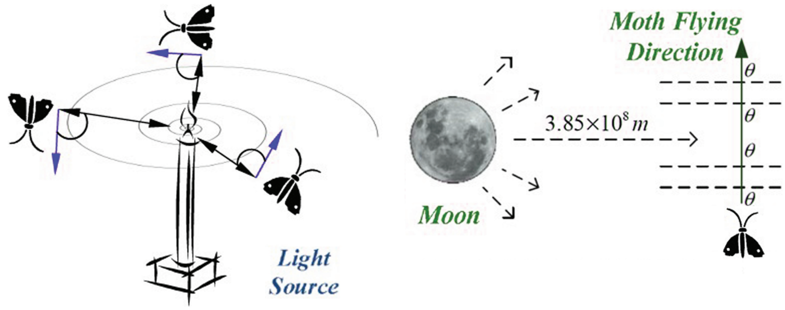
Moth Flame Optimization (MFO) is a recent SI algorithm that emulates the natural movement of moths [19]. Moths travel in a straight line by applying a transfer orientation mechanism. Maintaining the same angle is possible only when the source light is far away from such as the moonlight. However, when the source light is close such as the light of a candle, the moth is forced to move spirally. Fig.1 shows the conceptual model of the MFO algorithm. Eq.1 describes mathematically the natural spiral motion of moths around a flame where M_i represents the i_{th} moth, F_j represents the j_{th} flame, and S is the spiral function. Eq.2 formulates the spiral motion using a standard logarithmic function where D_i is the distance between the i_{th} moth and the j_{th} flame as described in Eq.3, b is a constant value for determining the shape of the logarithmic spiral, and t is a random number in the range [-1, 1]. The parameter $t = -1$ indicates the closest position of a moth to a flame where $t = 1$ indicates the farthest position between a moth and a flame. To achieve more exploitation in the search space the t parameter is considered in the range [r, 1] where r is linearly decreased throughout iterations from -1 to -2. Eq.4 shows gradual decrements of the number of flames throughout iterations where l is the current number of iteration, N is the maximum number of flames and T is the maximum number of iterations. Algorithm 1 shows the entire pseudo code of the MFO algorithm.

$$M_i = S(M_i, F_j) \tag{1}$$

$$S(M_i, F_j) = D_i \times e^{bt} \times cos(2\pi) + F_j \tag{2}$$

$$D_i = |M_i - F_j| \tag{3}$$

$$FlameNo = round(N - l \times (N - 1)/T) \tag{4}$$



(a) Moths fly in a spiral path into nearby light source.

(b) Moths flying in a fixed angle relative to moonlight.

Fig. 1: The conceptual model for the movement behaviour of moths

Algorithm 1 Pseudo-code of the MFO algorithm

Input:$Max\_iteration$, n (number of moths), d (number of dimensions)
Output:Approximated global solution
Initialize the position of moths
  while $l \leq Max\_iteration$ **do**
    Update flame no using Eq.4
    OM = FitnessFunction(M);
    if l == 1 **then**
      F = $sort(M)$;
      OF = $sort(OM)$;
    else
      F = $sort(M_{l-1}, M_l)$;
      OF = $sort(OM_{l-1}, OM_l)$;
    end if
    for i = 1: n **do**
      for j = 1: d **do**
        Update r and t;
        Calculate D using Eq.3 with respect to the corresponding moth;
        Update $M(i, j)$ using Eqs.1 and Eqs.2 with respect to the corresponding moth;
      end for
    end for
    l = l + 1;
  end while

B. Binary Moth Flame Optimisation

The original MFO was designed to deal with continuous search space in which the solution is composed of real values [19]. For the discrete search space, the solution is composed of binary values either "0" or "1." This implies that the MFO should be modified by integrating some operators that guarantee that this constraint on the solutions is not violated. The most common binary operator used to convert continuous optimizers into binary is the transfer function (TF) [20]. The main reason for using TFs is that they are easy to implement without affecting the concept of the algorithm. In this paper, the used TF is the sigmoid function which was used originally in [21] to generate the binary PSO (BPSO). In the MFO algorithm, the first term of Eq.2 represents the step vector which is redefined in Eq.5. The function of the sigmoid is to determine a probability value in the range [0,1] for each element of the solution. Eq.6 shows the formula of the sigmoid function. Each moth updates its position based on Eq.7 which takes the output of Eq.6 as its input.

$$\Delta\ M = D_i\ \times\ e^{bt}\ \times\ cos(2\pi) \tag{5}$$

$$TF(\Delta\ M_t) = 1/(1 + e^{\Delta\ M_t}) \tag{6}$$

$$M_i^d(t+1) = \begin{cases} 0, & \text{if } rand < TF(\Delta\ M_{t+1}) \\ 1, & \text{if } rand \geqslant TF(\Delta\ M_{t+1}) \end{cases} \tag{7}$$

III. THE PROPOSED APPROACH

This section presents the proposed approach by explaining the FS algorithm and the used evaluation criterion.

A. FS algorithm

In this paper, the BMFO algorithm is used as a search algorithm in the FS process. BMFO has been used effectively in various discrete problems such as FS. BMFO is similar to MFO in that the update process of a moth doesn't take into consideration the fitness value of a moth. This means that low-quality moths will change their positions in the search space as high-quality solutions. Ignoring the closeness of a moth from the optimal solution and updating them without caring about its fitness may degrade the performance of the optimizer. Therefore, this work addresses this issue by incorporating the rank of a moth in the update process. Thus, the original update strategy of a moth in the standard MFO algorithm that is formulated by Eq.2 will be modified to be an adaptive update strategy as illustrated in Eq.8. The added term is $\frac{R_i}{N}$ where R_i indicates the rank of the i moth in the swarm and N represents the size of the swarm.

$$S(M_i, F_j) = \left(D_i \times e^{bt} \times cos(2\pi) + F_j\right) \times \left(\frac{R_i}{N}\right) \quad (8)$$

In this strategy, each moth will be given a rank based on its fitness value. The high-quality moths with high fitness values will have a small rank and therefore there will be small changes in their positions. This enables the optimizer to search locally and do more exploitation for the specified region in the search space. This is useful when the moth is close to the optimal solution because it will increase the opportunity to reach global optima. In contrast, the low-quality moths with low fitness values will be given high ranks which forces them to violently change their positions and search globally to explore more regions in the feature space.

B. Evaluation criterion

The proposed FS approach applies the wrapper method to evaluate the candidate feature subset that represents a possible solution for the FS problem. Two important criteria must be involved in the fitness function when designing a wrapper FS algorithm: maximizing the performance of a learning algorithm (e.g. classification accuracy) and minimizing the number of selected features simultaneously.

Eq.9 formulates the FS problem where $\alpha\gamma_R(D)$ is the error rate of the classification produced by a classifier, $|R|$ is the number of selected features in the reduced data set, and $|C|$ is the number of features in the original data set, and $\alpha \in [0, 1]$, $\beta = (1 - \alpha)$ are two parameters for representing the importance of classification performance and length of feature subset based on recommendations [28].

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

IV. EXPERIMENTAL RESULTS

In this paper, sixteen medical data sets from well-regarded data repositories [38]–[40] were used to evaluate the modified wrapper approach. Table I shows the details of these data sets. Table II shows the parameters settings of three well-known meta-heuristic algorithms: BGWO, BCS, and BBA. These wrapper based approaches were used for comparison with the proposed approach. All the experiments were executed on a personal machine with AMD Athlon Dual-Core QL-60 CPU at 1.90 GHz and memory of two GB running Windows7 Ultimate 64 bit operating system. The optimization algorithms have been all implemented in Python in the EvoloPy-FS framework [1]. The maximum number of iterations and the population size were set to 100 and 10 respectively. In this work, the K-NN classifier (K = 5 [1]) is used to assess the goodness of each solution in the wrapper FS approach. Each data set is randomly divided into two parts; 66% for training and 34% for testing. To obtain statistically significant results, this division was repeated thirty independent and the final statistical results represent the average results of these runs. The α and β parameters in the fitness equation is set to 0.99 and 0.01, respectively [2].

TABLE I: Description of the used datasets

NO	Dataset Name	No features	No instances	No classes
1	Breast Cancer Wisconsin (Diagnostic)	30	569	2
2	Breast Cancer Wisconsin (Original)	9	699	2
3	Breast Cancer Coimbra	9	115	2
4	BreastEW	30	596	2
5	Dermatology	34	366	6
6	ILPD (Indian Liver Patient Dataset)	10	583	2
7	Lymphography	18	148	4
8	Parkinsons	22	194	2
9	SPECT	22	267	2
10	HeartEW	13	270	2
11	Hepatitis	18	79	2
12	South African Heart (SA Heart)	9	461	2
13	SPECTF Heart	43	266	2
14	Heart	13	302	5
15	Pima-indians-diabetes	9	768	2
16	Colon	2000	62	2
17	Leukemia	7129	72	2

TABLE II: Parameter settings.

Algorithm	Parameter	Value
GWO	α	[2,0]
BA	Qmin Frequency minimum	0
	Qmax Frequency maximum	2
	A Loudness	0.5
	r Pulse rate	0.5
CS	pa	0.25
	β	3/2

The results were analyzed in two steps: firstly, by comparing the standard BMFO algorithm and RBMFO together to study the effect of the rank-based modification strategy on the optimization ability of the algorithm in the feature space. The performance of the algorithms is evaluated in terms of classification accuracy, number of selected features, fitness values and running time evaluation measures. Secondly, involves comparing the proposed RBMFO with three well-regarded wrapper based approaches (e.g. BGWO, BCS, and BBA) using the same environment and using the same evaluation measures in the first step.

Inspecting the results in Table III, it can be observed that the results of BMFO and RBMFO are very competitive in terms of classification accuracy. RBMFO obtained the best results across nine of the sixteen data sets while BMFO scored the best results across the remaining seven data sets. Generally, both approaches have approximately the same classification performance because the difference between accuracy results is not significant. From the other side, the standard deviation for the accuracy results over thirty runs shows that RBMFO is more stable than BMFO.

Table IV reports the fitness values for both approaches. It can be seen that, in general, RBMFO achieved better fitness results compared to BMFO. RBMFO obtained the best

TABLE III: Comparison between BMFO and RBMFO in terms of the average classification accuracy

Dataset Name	BMFO		RBMFO	
	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	**0.907**	0.006	0.895	0.003
Breast Cancer Wisconsin (Original)	0.969	0.003	**0.971**	0.001
Breast Cancer Coimbra	0.775	0.000	**0.802**	0.000
BreastEW	**0.935**	0.011	0.927	0.008
Dermatology	**0.972**	0.028	0.952	0.012
ILPD (Indian Liver Patient Dataset)	0.648	0.000	**0.653**	0.023
Lymphography	0.673	0.059	**0.693**	0.029
Parkinsons	0.776	0.000	**0.807**	0.000
SPECT	0.657	0.029	**0.662**	0.019
HeartEW	**0.788**	0.022	0.778	0.019
South African Heart (SA Heart)	0.627	0.002	**0.649**	0.001
SPECTF Heart	**0.768**	0.027	0.763	0.021
Heart	0.741	0.037	**0.764**	0.028
Pima-indians-diabetes	**0.802**	0.000	0.725	0.000
Colon	**0.652**	0.051	0.632	0.048
Leukemia	0.852	0.029	**0.857**	0.028

results in nine data sets, while BMFO achieved the minimum fitness values for only four data sets. Fitness values remained the same for the ILPD (Indian Liver Patient Data set), Pima-Indians-diabetes and Leukemia data sets. Furthermore, RBMFO achieved more stability in fitness results when it runs for thirty times. Because the fitness function includes both classification accuracy and reduction rate, it can be inferred that the overall performance of RBMFO is better than BMFO.

TABLE IV: Comparison between BMFO and RBMFO in terms of the average fitness value

Dataset Name	BMFO		RBMFO	
	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	**0.038**	0.022	0.083	0.011
Breast Cancer Wisconsin (Original)	0.049	0.007	**0.042**	0.003
Breast Cancer Coimbra	0.287	0.073	**0.274**	0.000
BreastEW	**0.071**	0.025	0.076	0.021
Dermatology	**0.061**	0.040	0.084	0.024
ILPD (Indian Liver Patient Dataset)	**0.345**	0.000	**0.345**	0.017
Lymphography	**0.405**	0.077	0.409	0.029
Parkinsons	0.348	0.000	**0.347**	0.000
SPECT	0.371	0.050	**0.369**	0.036
HeartEW	0.196	0.061	**0.177**	0.047
South African Heart (SA Heart)	0.406	0.014	**0.404**	0.010
SPECTF Heart	0.303	0.066	**0.263**	0.050
Heart	0.243	0.075	**0.229**	0.068
Pima-indians-diabetes	**0.258**	0.000	**0.258**	0.000
Colon	0.377	0.000	**0.376**	0.000
Leukemia	**0.115**	0.000	**0.115**	0.000

From Table V, it can be seen clearly that RBMFO outperformed BMFO in reducing the number of features in the specified feature subset. RBMFO recorded the minimum number of selected features across 75% of the data sets. Sure, this is very useful especially for large medical data sets as the goal of the FS process is to create the smallest size feature subset with the most useful features. Moreover, RBMFO showed more stability in results when the experiments were repeated thirty times.

Table VI shows the average running time results for both approaches. It is clearly shown that the RBMFO requires the smallest running time to converge toward the global solution. RBMFO obtained the smallest running time in eleven of the sixteen data sets while BMFO had the smallest running time on only five data sets. Furthermore, the RBMFO shows more stability in the running time results when the experiments were repeated thirty times.

TABLE V: Comparison between BMFO and RBMFO in terms of the average number of the selected features

Dataset Name	BMFO		RBMFO	
	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	16.500	1.367	**11.833**	1.075
Breast Cancer Wisconsin (Original)	4.400	0.932	**3.900**	0.305
Breast Cancer Coimbra	**4.000**	0.254	4.067	0.000
BreastEW	19.433	1.888	**14.200**	1.789
Dermatology	22.167	2.849	**17.467**	1.877
ILPD (Indian Liver Patient Dataset)	**3.967**	0.183	4.000	0.000
Lymphography	9.367	1.591	**7.233**	0.718
Parkinsons	8.233	0.776	**6.467**	1.040
SPECT	14.167	2.401	**10.600**	1.464
HeartEW	7.767	1.569	**6.767**	0.817
South African Heart (SA Heart)	**4.030**	0.305	4.100	0.183
SPECTF Heart	29.500	3.192	**21.133**	3.048
Heart	5.700	0.973	**4.867**	0.596
Pima-indians-diabetes	**4.000**	0.000	4.033	0.183
Colon	1082.067	164.537	**969.267**	15.733
Leukemia	3506.367	21.886	**3435.467**	17.122

TABLE VI: Comparison between BMFO and RBMFO in terms of the average running time (seconds)

Dataset Name	BMFO		RBMFO	
	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	21.476	0.537	**20.351**	0.100
Breast Cancer Wisconsin (Original)	23.241	0.405	**22.108**	0.105
Breast Cancer Coimbra	7.918	0.368	**7.878**	0.074
BreastEW	23.493	0.566	**21.736**	0.094
Dermatology	16.038	0.619	**15.385**	0.091
ILPD (Indian Liver Patient Dataset)	13.724	0.469	**13.344**	0.058
Lymphography	**9.146**	0.608	9.222	0.135
Parkinsons	10.572	0.154	**10.339**	0.079
SPECT	12.418	0.314	**12.349**	0.072
HeartEW	**11.848**	0.527	11.907	0.086
South African Heart (SA Heart)	**16.354**	0.640	16.381	0.095
SPECTF Heart	13.866	0.580	**13.546**	0.090
Heart	12.165	0.224	**11.812**	0.063
Pima-indians-diabetes	24.727	0.543	**23.845**	0.128
Colon	**76.932**	1.044	80.271	0.854
Leukemia	**265.892**	3.009	276.099	5.121

The second comparison was performed between the proposed approach and three well-regarded approaches using the same evaluation measures and data sets. Observing the results in Table VII and Fig 2, it can be seen that RBMFO obtained the best accuracy results in nine out of sixteen data sets. BCS and BGWO achieved the best classification performance on five data sets while BBA did not score the best performance for any of the data sets. Moreover, RBMFO shows more stable results compared to other approaches. From the results of classification performance, it can be revealed that the embedded spiral operator of the MFO algorithm and the enhanced update strategy using the rank-based method have a significant impact on increasing the optimization ability of the algorithm and achieving a better trade-off between exploration and exploitation phases. This is because the RBMFO will alternate effectively between exploration and exploitation phases. This is accomplished by emphasizing more exploitation around the high-quality solution and stressing further exploration of the search space in the event of low-quality solutions.

TABLE VII: Comparison between RBMFO and other approaches in terms of the average classification accuracy

Dataset Name	RBMFO		BGWO		BCS		BBA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	0.895	0.003	**0.904**	0.015	0.896	0.004	0.760	0.065
Breast Cancer Wisconsin (Original)	**0.971**	0.001	0.963	0.012	**0.971**	0.001	0.939	0.022
Breast Cancer Coimbra	**0.802**	0.000	0.707	0.052	0.777	0.139	0.583	0.037
BreastEW	**0.927**	0.008	0.925	0.013	0.923	0.011	0.896	0.017
Dermatology	0.952	0.012	0.956	0.021	**0.961**	0.020	0.739	0.069
ILPD (Indian Liver Patient Dataset)	**0.653**	0.023	0.591	0.066	**0.653**	0.000	0.582	0.049
Lymphography	0.693	0.029	**0.956**	0.044	0.690	0.051	0.705	0.066
Parkinsons	**0.807**	0.000	0.799	0.037	0.776	0.000	0.786	0.037
SPECT	**0.662**	0.019	0.656	0.036	0.653	0.019	0.624	0.052
HeartEW	**0.778**	0.019	0.770	0.028	**0.778**	0.020	0.709	0.051
South African Heart (SA Heart)	**0.649**	0.001	0.647	0.022	0.627	0.002	0.622	0.033
SPECTF Heart	0.763	0.021	**0.767**	0.032	0.760	0.032	0.748	0.035
Heart	0.764	0.028	0.723	0.041	**0.769**	0.140	0.645	0.063
Pima-indians-diabetes	0.725	0.000	**0.750**	0.018	0.725	0.000	0.664	0.046
Colon	0.632	0.048	**0.655**	0.047	0.632	0.053	0.645	0.051
Leukemia	**0.857**	0.028	0.856	0.023	0.853	0.032	0.715	0.027

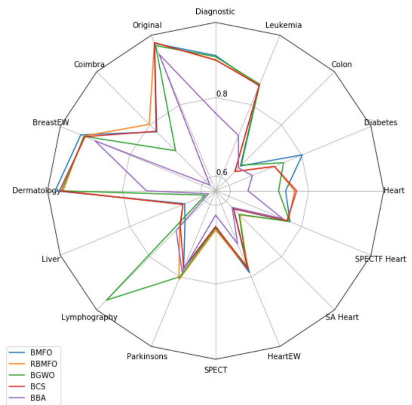


Fig. 2: Accuracy based comparison

Regarding fitness value results, Table VIII and Fig 3 show that RBMFO and BCS have achieved competitive results in terms of fitness values. However, RBMFO has more stability in results. BGWO came next and was the best across five data sets while BBA achieved the best in only three data sets.

TABLE VIII: Comparison between RBMFO and other approaches in terms of the average fitness value

Dataset Name	RBMFO		BGWO		BCS		BBA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	0.083	0.011	**0.074**	0.0137	0.085	0.018	0.322	0.084
Breast Cancer Wisconsin (Original)	**0.042**	0.003	0.069	0.023	**0.042**	0.094	0.079	0.035
Breast Cancer Coimbra	**0.274**	0.000	0.276	0.140	0.287	0.059	0.557	0.029
BreastEW	0.076	0.021	0.075	0.022	**0.068**	0.025	0.093	0.028
Dermatology	0.084	0.024	0.090	0.032	**0.070**	0.034	0.436	0.104
ILPD (Indian Liver Patient Dataset)	**0.345**	0.017	0.407	0.060	**0.345**	0.000	0.414	0.054
Lymphography	0.409	0.029	**0.405**	0.066	0.433	0.094	0.431	0.065
Parkinsons	0.347	0.000	**0.329**	0.067	0.347	0.001	0.366	0.057
SPECT	0.369	0.036	0.358	0.056	0.374	0.046	**0.351**	0.041
HeartEW	0.177	0.047	0.194	0.048	**0.154**	0.053	0.290	0.080
South African Heart (SA Heart)	0.404	0.010	0.400	0.052	0.404	0.014	**0.371**	0.047
SPECTF Heart	**0.263**	0.050	0.276	0.061	0.264	0.059	0.272	0.051
Heart	0.229	0.068	0.278	0.075	**0.182**	0.078	0.456	0.045
Pima-indians-diabetes	**0.258**	0.000	0.261	0.039	**0.258**	0.000	0.386	0.061
Colon	**0.376**	0.000	**0.376**	0.000	**0.376**	0.000	**0.376**	0.000
Leukemia	**0.115**	0.000	**0.115**	0.000	0.118	0.020	0.329	0.000

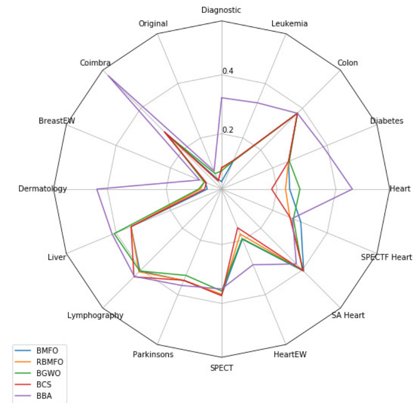


Fig. 3: Fitness based comparison

By examining Table IX, it appears that BBA was the best approach to reduce the number of selected features. BBA was superior in ten data sets. BCS and RBMFO were the best across four and three data sets respectively. On the other hand, BGWO did not achieve the best feature reduction ratio over any of the data sets.

TABLE IX: Comparison between RBMFO and other approaches in terms of the average number of selected features

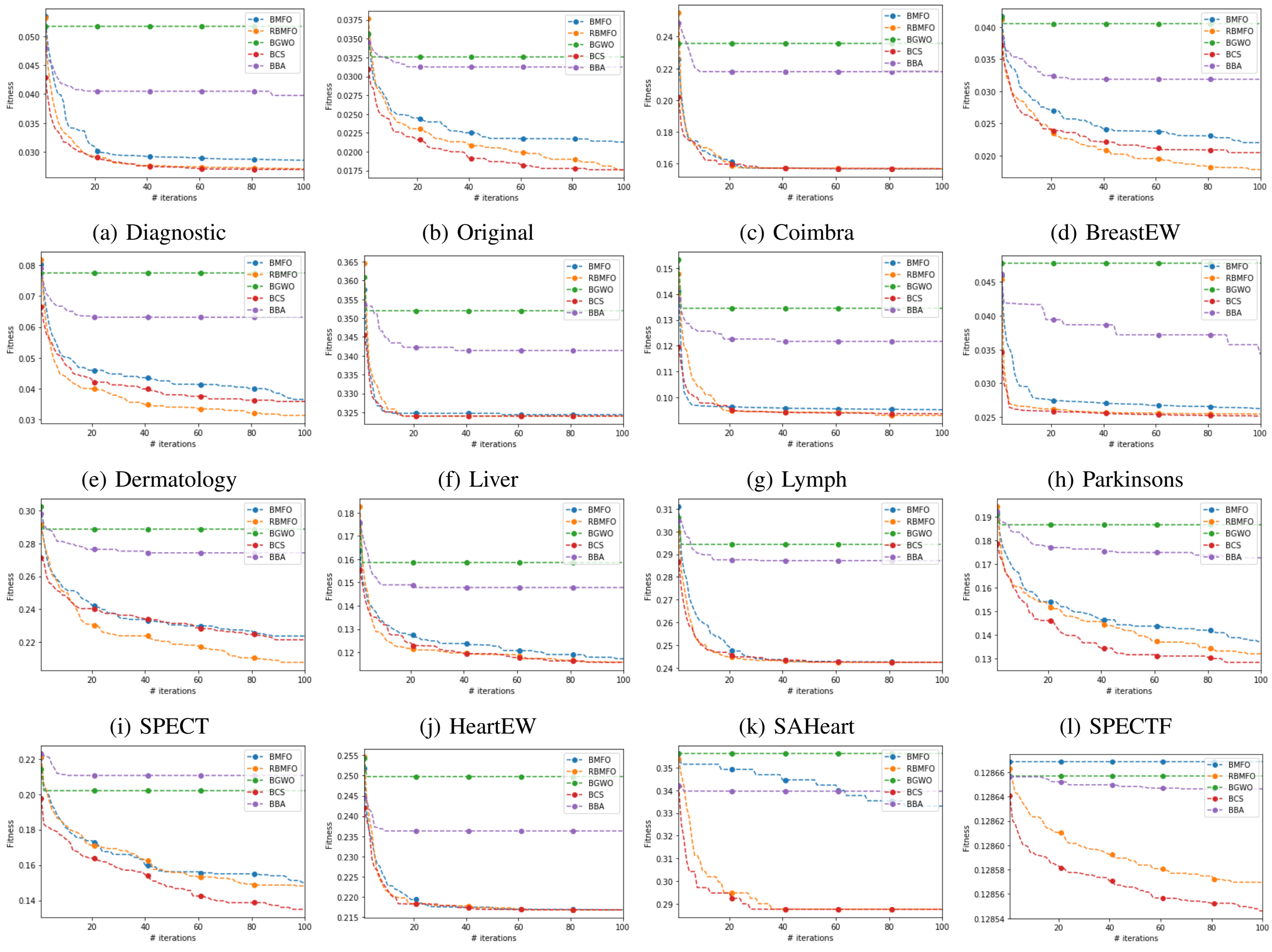
Dataset Name	RBMFO		BGWO		BCS		BBA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	11.833	1.075	14.733	3.062	**11.533**	1.756	14.533	3.421
Breast Cancer Wisconsin (Original)	3.900	0.305	5.667	1.295	3.933	0.254	**3.633**	1.564
Breast Cancer Coimbra	**4.067**	0.000	4.333	1.398	4.133	1.974	4.600	1.192
BreastEW	14.200	1.789	15.600	1.567	14.933	2.348	**12.500**	2.224
Dermatology	17.467	1.877	17.267	2.392	16.667	2.202	**14.367**	2.785
ILPD (Indian Liver Patient Dataset)	4.000	0.000	3.533	0.730	4.000	0.000	**2.467**	1.106
Lymphography	7.233	0.718	9.767	2.176	7.000	1.875	**6.600**	2.513
Parkinsons	6.467	1.040	9.333	1.826	**5.800**	1.186	10.400	2.268
SPECT	10.600	1.464	10.667	2.249	10.400	2.207	**9.000**	2.767
HeartEW	6.767	0.817	7.100	1.561	6.200	1.769	**5.467**	1.9429
South African Heart (SA Heart)	**4.100**	0.183	4.433	1.223	**4.100**	0.305	4.167	1.416
SPECTF Heart	21.133	3.048	22.033	3.882	19.933	3.194	**17.900**	4.641
Heart	**4.867**	0.596	4.767	1.194	5.467	0.945	5.967	2.059
Pima-indians-diabetes	4.033	0.183	4.033	0.999	4.033	0.183	**3.767**	1.431
Colon	969.267	15.733	970.867	14.219	**961.867**	13.475	974.233	65.520
Leukemia	3435.467	17.122	3495.567	27.931	3422.367	18.524	**1241.767**	53.723

To analyze optimization time for a different approach, Table X records the run time for each algorithm. It is noted that the proposed approach achieved the minimum run time to reach the convergence state. This can be seen in 88% of data sets. On the other hand, the BBA had the smallest run time on only two data sets, Colon and Leukemia. BGWO and BCS were not the best run time for any data set.

TABLE X: Comparison between RBMFO and other approaches in terms of the average running time (seconds)

Dataset Name	RBMFO		BGWO		BCS		BBA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Breast Cancer Wisconsin (Diagnostic)	**20.351**	0.100	23.426	0.941	41.084	0.558	21.125	0.282
Breast Cancer Wisconsin (Original)	**22.108**	0.105	24.175	0.849	45.724	0.608	23.015	0.298
Breast Cancer Coimbra	**7.878**	0.074	8.690	0.353	15.852	5.578	8.274	0.131
BreastEW	**21.736**	0.094	24.955	1.068	43.112	1.020	22.147	0.418
Dermatology	**15.385**	0.091	18.405	0.731	30.140	0.551	15.607	0.221
ILPD (Indian Liver Patient Dataset)	**13.344**	0.058	14.249	0.714	27.131	0.407	14.001	0.173
Lymphography	**9.146**	0.135	10.775	0.201	17.249	0.369	9.206	0.106
Parkinsons	**10.339**	0.079	13.064	0.747	20.188	0.300	10.628	0.145
SPECT	**12.349**	0.072	14.483	0.607	24.256	0.358	12.674	1.434
HeartEW	**11.907**	0.086	13.625	0.632	23.590	0.278	12.395	0.151
South African Heart (SA Heart)	**16.381**	0.095	17.198	0.476	33.451	0.469	17.170	0.256
SPECTF Heart	**13.546**	0.090	18.002	0.437	25.759	0.351	13.597	0.172
Heart	**11.812**	0.063	13.562	0.757	23.625	4.153	12.324	0.142
Pima-indians-diabetes	**23.845**	0.128	25.043	0.879	49.313	0.778	24.964	0.389
Colon	80.271	0.854	274.127	3.479	73.273	1.254	**55.723**	0.673
Leukemia	276.099	5.121	964.913	5.703	245.912	4.022	**189.204**	3.002

Fig 4 illustrates the convergence behavior of all the studied wrapper approaches on all data sets. Each subfigure shows the changes in fitness value for each approach across all



(a) Diagnostic (b) Original (c) Coimbra (d) BreastEW

(e) Dermatology (f) Liver (g) Lymph (h) Parkinsons

(i) SPECT (j) HeartEW (k) SAHeart (l) SPECTF

(m) Heart (n) Diabetes (o) Colon (p) Leukemia

Fig. 4: Convergence curves for the rank-based BMFO and other wrapper-based approaches on the used medical data sets

iterations on a specified data set. In all data sets, RBMFO and BCS show the best convergence trends compared to other approaches. This can be realized from the convergence curves that achieve the minimum fitness values in the final iterations of the optimization process. On the other hand, premature convergence and entrapment in local minima can be guessed from the convergence behavior of BGWO and BBA wrapper approaches.

V. CONCLUSIONS

This paper proposes a new wrapper-based FS approach to improve the classification tasks in the medical application. The MFO is used as a search algorithm in the FS process and K-NN as an evaluator to decide the quality of the generated feature subset. The main contribution of this work is to enhance the optimization capability of the BMFO in the feature space. The proposed approach adopts a ranked-based update strategy that uses the fitness value of an individual to adaptively update its position. The RBMFO is tested on sixteen benchmark medical data sets from well-regarded data repositories. Then, it is compared with three wrapper approaches that are tested on the same data sets. The experimental results show that the rank-based updating strategy performs better than the standard update strategy and better than the other approaches. For future direction, we would like to apply the proposed approach to other challenging optimization problems.

REFERENCES

[1] Khurma, R. A., Aljarah, I., Sharieh, A., Mirjalili, S. (2020). EvoloPy-FS: An Open-Source Nature-Inspired Optimization Framework in Python for Feature Selection. In Evolutionary Machine Learning Techniques (pp. 131-173). Springer, Singapore.

[2] Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., Mirjalili, S. (2020). Efficient hybrid nature-inspired binary optimizers for feature selection. Cognitive Computation, 12(1), 150-175.

[3] Brezočnik, Lucija, Iztok Fister, and Vili Podgorelec. "Swarm intelligence algorithms for feature selection: a review." Applied Sciences 8.9 (2018): 1521.

[4] Mistry, Kamlesh, et al. "A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition." IEEE transactions on cybernetics 47.6 (2016): 1496-1509.

[5] Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A. M., Castillo, P. A., Aljarah, I. (2020). Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market. Progress in Artificial Intelligence, 9(1), 31-53.

[6] Al-Madi, N., Faris, H., Abukhurma, R. (2018). Cost-Sensitive Genetic Programming for Churn Prediction and Identification of the Influencing Factors in Telecommunication Market. International Journal of Advanced Science and Technology, 13-28."

[7] Tubishat, M., Abushariah, M. A., Idris, N., Aljarah, I. (2019). Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. Applied Intelligence, 49(5), 1688-1707.

[8] Ghamisi, Pedram, Micael S. Couceiro, and Jon Atli Benediktsson. "A novel feature selection approach based on FODPSO and SVM." IEEE Transactions on Geoscience and Remote Sensing 53.5 (2014): 2935-2947.

[9] Mandal, Monalisa, Anirban Mukhopadhyay, and Ujjwal Maulik. "Prediction of protein subcellular localization by incorporating multiobjective PSO-based feature subset selection into the general form of Chou's PseAAC." Medical & biological engineering & computing 53.4 (2015): 331-344.

[10] Raymer, Michael L., et al. "Dimensionality reduction using genetic algorithms." IEEE transactions on evolutionary computation 4.2 (2000): 164-171.

[11] Guo, Jianmei, et al. "A genetic algorithm for optimized feature selection with resource constraints in software product lines." Journal of Systems and Software 84.12 (2011): 2208-2221.

[12] Zhang, Yudong, et al. "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection." Knowledge-Based Systems 64 (2014): 22-31.

[13] Wang, Mingjing, et al. "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses." Neurocomputing 267 (2017): 69-84.

[14] Wang, Mingjing, et al. "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses." Neurocomputing 267 (2017): 69-84.

[15] Inbarani, H. Hannah, Ahmad Taher Azar, and G. Jothi. "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis." Computer methods and programs in biomedicine 113.1 (2014): 175-185.

[16] Zhang, Li, et al. "Intelligent facial emotion recognition using moth-firefly optimization." Knowledge-Based Systems 111 (2016): 248-267.

[17] Zawbaa, Hossam M., et al. "Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach." Swarm and Evolutionary Computation 42 (2018): 29-42.

[18] Medjahed, Seyyid Ahmed, et al. "Gray wolf optimizer for hyperspectral band selection." Applied Soft Computing 40 (2016): 178-186.

[19] Mirjalili, Seyedali. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." Knowledge-Based Systems 89 (2015): 228-249.

[20] Mirjalili, Seyedali, and Andrew Lewis. "S-shaped versus V-shaped transfer functions for binary particle swarm optimization." Swarm and Evolutionary Computation 9 (2013): 1-14.

[21] Kennedy, James, and Russell C. Eberhart. "A discrete binary version of the particle swarm algorithm." 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation. Vol. 5. IEEE, 1997.

[22] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1. No. 10. New York: Springer series in statistics, 2001.

[23] Yuhui Shi and Russell C Eberhart. 1999. Empirical study of particle swarm optimization. In Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on, Vol. 3. IEEE, 1945–1950.

[24] Yang, Chengwei, et al. "Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight." Applied Soft Computing 29 (2015): 386-394.

[25] Fan, Shu-Kai S., and Yi-Yin Chiu. "A decreasing inertia weight particle swarm optimizer." Engineering Optimization 39.2 (2007): 203-228.

[26] Kentzoglanakis, Kyriakos, and Matthew Poole. "Particle swarm optimization with an oscillating inertia weight." GECCO'09-11th Annual conference on Genetic and evolutionary computation. 2009.

[27] Gao, Yue-lin, Xiao-hui An, and Jun-min Liu. "A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation." 2008 International Conference on Computational Intelligence and Security. Vol. 1. IEEE, 2008.

[28] Mafarja, M., Aljarah, I., Heidari, A. A., Faris, H., Fournier-Viger, P., Li, X., Mirjalili, S. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowledge-Based Systems, 161, 185-204.

[29] Sayed, Gehad Ismail, and Aboul Ella Hassanien. "A hybrid SA-MFO algorithm for function optimization and engineering design problems." Complex & Intelligent Systems 4.3 (2018): 195-212.

[30] Sayed, Gehad Ismail, et al. "Alzheimer's disease diagnosis based on moth flame optimization." International Conference on Genetic and Evolutionary Computing. Springer, Cham, 2016.

[31] Wang, Mingjing, et al. "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses." Neurocomputing 267 (2017): 69-84.

[32] R. Y. Nakamura, L. A. Pereira, K. Costa, D. Rodrigues, J. P. Papa, X.-S. Yang, Bba: a binary 3318 bat algorithm for feature selection, in: 2012 25th SIBGRAPI conference on graphics, Patterns 3319 and Images, IEEE, pp. 291–297

[33] Rodrigues, Douglas, et al. "BCS: A binary cuckoo search algorithm for feature selection." 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013). IEEE, 2013.

[34] Khushaba, Rami N., Ahmed Al-Ani, and Adel Al-Jumaily. "Feature subset selection using differential evolution and a statistical repair mechanism." Expert Systems with Applications 38.9 (2011): 11515-11526.

[35] Reddy, Srikanth, et al. "Solution to unit commitment in power system operation planning using binary coded modified moth flame optimization algorithm (BMMFOA): a flame selection based computational technique." Journal of Computational Science 25 (2018): 298-317.

[36] Mehne, Seyed Hamed Hashemi, and Seyedali Mirjalili. "Moth-flame optimization algorithm: theory, literature review, and application in optimal nonlinear feedback control design." Nature-Inspired Optimizers. Springer, Cham, 2020. 143-166.

[37] Shehab, Mohammad, et al. "Moth–flame optimization algorithm: variants and applications." Neural Computing and Applications (2019): 1-26.

[38] Asuncion, Arthur, and David Newman. "UCI machine learning repository." (2007).

[39] Alcalá-Fdez, Jesús, et al. "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." Journal of Multiple-Valued Logic and Soft Computing 17 (2011).

[40] Goldbloom, A., et al. "Kaggle: your homr for data science." (2017).

[41] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in engineering software 69 (2014): 46-61.

[42] Tang, Jiliang, Salem Alelyani, and Huan Liu. "Feature selection for classification: A review." Data classification: Algorithms and applications (2014): 37.

[43] Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.

[44] Khurma, R.; Aljarah, I. and Sharieh, A. (2020). An Efficient Moth Flame Optimization Algorithm using Chaotic Maps for Feature Selection in the Medical Applications.In Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, ISBN 978-989-758-397-1, ISSN 2184-4313, pages 175-182. DOI: 10.5220/0008960701750182

[45] Faris, H., Aljarah, I., Mirjalili, S. (2018). Improved monarch butterfly optimization for unconstrained global search and neural network training. Applied Intelligence, 48(2), 445-464.

[46] Taradeh, M., Mafarja, M., Heidari, A. A., Faris, H., Aljarah, I., Mirjalili, S., Fujita, H. (2019). An evolutionary gravitational search-based feature selection. Information Sciences, 497, 219-239.

[47] Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., Mirjalili, S. (2020). Efficient hybrid nature-inspired binary optimizers for feature selection. Cognitive Computation, 12(1), 150-175.

[48] Faris, H., Aljarah, I., Alqatawna, J. F. (2015, November). Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection. In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT) (pp. 1-5). IEEE.

[49] Faris, H., Aljarah, I., Mirjalili, S., Castillo, P. A., Guervós, J. J. M. (2016, November). EvoloPy: An Open-source Nature-inspired Optimization Framework in Python. In IJCCI (ECTA) (pp. 171-177).

[50] Al-Betar, M. A., Awadallah, M. A., Faris, H., Aljarah, I., Hammouri, A. I. (2018). Natural selection methods for grey wolf optimizer. Expert Systems with Applications, 113, 481-498.