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
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.

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

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

\[
D_i = |M_i - F_j| \tag{3}
\]

\[
FlameNo = round(N - l \times (N - 1)/T) \tag{4}
\]
Algorithm 1 Pseudo-code of the MFO algorithm

Input: \(\text{Max}\_\text{iteration}, \ n\) (number of moths), \(d\) (number of dimensions)
Output: Approximated global solution

Initialize the position of moths

\[
\text{while } l \leq \text{Max\_iteration} \ \text{do}
\]
Update flame no using Eq.4
\[
\text{OM} = \text{FitnessFunction}(M);
\]
if \(l == 1\) then
\[
\begin{align*}
F &= \text{sort}(M); \\
OF &= \text{sort}(OM);
\end{align*}
\]
else
\[
\begin{align*}
F &= \text{sort}(M_{l-1}, M_l); \\
OF &= \text{sort}(OM_{l-1}, OM_l);
\end{align*}
\]
end if

\[
\text{for } i = 1: n \ \text{do}
\]
\[
\text{for } j = 1: d \ \text{do}
\]
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;
\[
\text{end for}
\]
\[
\text{end for}
\]
\[
l = l + 1;
\]
\[
\text{end while}
\]

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.

\[
\Delta M = D_i \times e^{bt} \times \cos(2\pi)
\]

\[
TF(\Delta M) = \frac{1}{1 + e^{\Delta M}}
\]

\[
M_d(t + 1) = \begin{cases} 
0, & \text{if } \text{rand} < TF(\Delta M_{t+1}) \\
1, & \text{if } \text{rand} \geq TF(\Delta M_{t+1})
\end{cases}
\]

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].

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

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 $\alpha$ and $\beta$ parameters in the fitness equation is set to 0.99 and 0.01, respectively [2].

<table>
<thead>
<tr>
<th>NO</th>
<th>Dataset Name</th>
<th>No features</th>
<th>No instances</th>
<th>No classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Breast Cancer Wisconsin (Diagnostic)</td>
<td>30</td>
<td>589</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Breast Cancer Wisconsin (Original)</td>
<td>9</td>
<td>699</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Breast Cancer Cochima</td>
<td>9</td>
<td>115</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>BreastEW</td>
<td>30</td>
<td>596</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Dermatology</td>
<td>34</td>
<td>366</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>BLF (Indian Liver Patient Dataset)</td>
<td>10</td>
<td>583</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Lymphography</td>
<td>18</td>
<td>148</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>Parkinson</td>
<td>22</td>
<td>194</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>SPECT</td>
<td>22</td>
<td>267</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>HeartEW</td>
<td>15</td>
<td>270</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Hepatitis</td>
<td>18</td>
<td>79</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>South African Heart (SA Heart )</td>
<td>9</td>
<td>481</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>SPECTIF Heart</td>
<td>43</td>
<td>266</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Heart</td>
<td>13</td>
<td>302</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>Pima-indians-diabetes</td>
<td>9</td>
<td>768</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>Colon</td>
<td>2000</td>
<td>62</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>Leukemia</td>
<td>7129</td>
<td>72</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GWO</td>
<td>$\alpha$</td>
<td>0.2</td>
</tr>
<tr>
<td>BA</td>
<td>Qmin Frequency minimum</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Qmax Frequency maximum</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$r$ Pulse rate</td>
<td>0.5</td>
</tr>
<tr>
<td>CS</td>
<td>$p_o$</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>$\beta$</td>
<td>0.02</td>
</tr>
</tbody>
</table>

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
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.

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.
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>
<thead>
<tr>
<th>Dataset Name</th>
<th>RBMFO Avg</th>
<th>RBMFO Std</th>
<th>BGWO Avg</th>
<th>BGWO Std</th>
<th>BCS Avg</th>
<th>BCS Std</th>
<th>BBA Avg</th>
<th>BBA Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer Wisconsin (Diagnostic)</td>
<td>0.018</td>
<td>0.011</td>
<td>0.084</td>
<td>0.015</td>
<td>0.230</td>
<td>0.018</td>
<td>0.322</td>
<td>0.014</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin (Original)</td>
<td>0.042</td>
<td>0.003</td>
<td>0.069</td>
<td>0.023</td>
<td>0.042</td>
<td>0.004</td>
<td>0.079</td>
<td>0.035</td>
</tr>
<tr>
<td>Breast Cancer Coimbra</td>
<td>0.274</td>
<td>0.000</td>
<td>0.276</td>
<td>0.140</td>
<td>0.287</td>
<td>0.059</td>
<td>0.557</td>
<td>0.029</td>
</tr>
<tr>
<td>BreastEW</td>
<td>0.076</td>
<td>0.021</td>
<td>0.075</td>
<td>0.022</td>
<td>0.068</td>
<td>0.025</td>
<td>0.093</td>
<td>0.028</td>
</tr>
<tr>
<td>Dermatology</td>
<td>0.084</td>
<td>0.042</td>
<td>0.080</td>
<td>0.032</td>
<td>0.070</td>
<td>0.034</td>
<td>0.145</td>
<td>0.054</td>
</tr>
<tr>
<td>ILPD (Indian Liver Patient Dataset)</td>
<td>0.345</td>
<td>0.017</td>
<td>0.347</td>
<td>0.066</td>
<td>0.345</td>
<td>0.000</td>
<td>0.414</td>
<td>0.054</td>
</tr>
<tr>
<td>Lymphography</td>
<td>0.490</td>
<td>0.029</td>
<td>0.490</td>
<td>0.066</td>
<td>0.433</td>
<td>0.039</td>
<td>0.431</td>
<td>0.065</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>0.347</td>
<td>0.030</td>
<td>0.329</td>
<td>0.097</td>
<td>0.347</td>
<td>0.001</td>
<td>0.367</td>
<td>0.057</td>
</tr>
<tr>
<td>SPECT</td>
<td>0.390</td>
<td>0.066</td>
<td>0.356</td>
<td>0.076</td>
<td>0.374</td>
<td>0.046</td>
<td>0.351</td>
<td>0.041</td>
</tr>
<tr>
<td>South African Heart (SA Heart)</td>
<td>0.410</td>
<td>0.016</td>
<td>0.430</td>
<td>0.052</td>
<td>0.400</td>
<td>0.044</td>
<td>0.391</td>
<td>0.078</td>
</tr>
<tr>
<td>SPECT Heart</td>
<td>0.263</td>
<td>0.050</td>
<td>0.276</td>
<td>0.061</td>
<td>0.264</td>
<td>0.059</td>
<td>0.272</td>
<td>0.051</td>
</tr>
<tr>
<td>Heart</td>
<td>0.229</td>
<td>0.068</td>
<td>0.278</td>
<td>0.175</td>
<td>0.287</td>
<td>0.078</td>
<td>0.456</td>
<td>0.045</td>
</tr>
<tr>
<td>Pima-indians-diabetes</td>
<td>0.258</td>
<td>0.050</td>
<td>0.261</td>
<td>0.039</td>
<td>0.258</td>
<td>0.000</td>
<td>0.386</td>
<td>0.061</td>
</tr>
<tr>
<td>Colon</td>
<td>0.376</td>
<td>0.000</td>
<td>0.376</td>
<td>0.000</td>
<td>0.376</td>
<td>0.000</td>
<td>0.376</td>
<td>0.000</td>
</tr>
<tr>
<td>Leukemia</td>
<td>0.118</td>
<td>0.015</td>
<td>0.115</td>
<td>0.000</td>
<td>0.118</td>
<td>0.018</td>
<td>0.129</td>
<td>0.000</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Dataset Name</th>
<th>RBMFO Avg</th>
<th>RBMFO Std</th>
<th>BGWO Avg</th>
<th>BGWO Std</th>
<th>BCS Avg</th>
<th>BCS Std</th>
<th>BBA Avg</th>
<th>BBA Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast Cancer Wisconsin (Diagnostic)</td>
<td>20.591</td>
<td>0.010</td>
<td>21.477</td>
<td>0.059</td>
<td>21.446</td>
<td>0.008</td>
<td>21.456</td>
<td>0.012</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin (Original)</td>
<td>7.287</td>
<td>0.074</td>
<td>6.999</td>
<td>0.553</td>
<td>7.167</td>
<td>0.578</td>
<td>7.274</td>
<td>0.313</td>
</tr>
<tr>
<td>Breast Cancer Coimbra</td>
<td>22.376</td>
<td>0.046</td>
<td>23.340</td>
<td>0.868</td>
<td>23.132</td>
<td>0.030</td>
<td>23.147</td>
<td>0.048</td>
</tr>
<tr>
<td>BreastEW</td>
<td>7.213</td>
<td>0.075</td>
<td>7.967</td>
<td>1.215</td>
<td>7.285</td>
<td>1.475</td>
<td>6.876</td>
<td>2.515</td>
</tr>
<tr>
<td>Dermatology</td>
<td>4.236</td>
<td>0.246</td>
<td>4.157</td>
<td>0.126</td>
<td>4.163</td>
<td>0.123</td>
<td>4.143</td>
<td>0.126</td>
</tr>
<tr>
<td>ILPD (Indian Liver Patient Dataset)</td>
<td>8.259</td>
<td>1.064</td>
<td>10.667</td>
<td>2.462</td>
<td>10.489</td>
<td>2.207</td>
<td>9.848</td>
<td>2.157</td>
</tr>
<tr>
<td>Lymphography</td>
<td>3.000</td>
<td>0.000</td>
<td>3.533</td>
<td>0.778</td>
<td>3.000</td>
<td>0.000</td>
<td>2.687</td>
<td>1.160</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>31.153</td>
<td>0.046</td>
<td>22.033</td>
<td>0.882</td>
<td>31.013</td>
<td>0.316</td>
<td>27.963</td>
<td>1.640</td>
</tr>
<tr>
<td>Heart</td>
<td>4.569</td>
<td>0.967</td>
<td>4.763</td>
<td>1.194</td>
<td>4.967</td>
<td>0.645</td>
<td>3.957</td>
<td>2.899</td>
</tr>
<tr>
<td>Pima-indians-diabetes</td>
<td>4.003</td>
<td>0.183</td>
<td>4.633</td>
<td>0.898</td>
<td>4.833</td>
<td>0.033</td>
<td>3.567</td>
<td>1.091</td>
</tr>
<tr>
<td>Colon</td>
<td>569.257</td>
<td>17.735</td>
<td>570.567</td>
<td>14.219</td>
<td>561.067</td>
<td>13.478</td>
<td>575.232</td>
<td>15.650</td>
</tr>
</tbody>
</table>

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
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


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


