

Neighborhood search methods with moth optimization algorithm as a wrapper method for feature selection problems

Malek Alzaqebah¹, Nashat Alrefai², Eman A. E. Ahmed³, Sana Jawarneh⁴, Mutasem K. Alsmadi⁵

^{1,2,3}Department of Mathematics, College of Science, Imam Abdulrahman Bin Faisal University, Saudi Arabia

^{1,2,3}Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Saudi Arabia

⁴Computer Science Department, Community College Dammam, Imam Abdulrahman Bin Faisal University, Saudi Arabia

⁵Department of MIS, College of Applied Studies and Community Service,
Imam Abdulrahman Bin Faisal University, Saudi Arabia

Article Info

Article history:

Received Oct 14, 2019

Revised Jan 27, 2020

Accepted Feb 8, 2020

Keywords:

Classification

Feature selection

Metaheuristic optimization

Moth optimization

Neighborhood search

ABSTRACT

Feature selection methods are used to select a subset of features from data, therefore only the useful information can be mined from the samples to get better accuracy and improves the computational efficiency of the learning model. Moth-flame Optimization (MFO) algorithm is a population-based approach, that simulates the behavior of real moth in nature, one drawback of the MFO algorithm is that the solutions move toward the best solution, and it easily can be stuck in local optima as we investigated in this paper, therefore, we proposed a MFO Algorithm combined with a neighborhood search method for feature selection problems, in order to avoid the MFO algorithm getting trapped in a local optima, and helps in avoiding the premature convergence, the neighborhood search method is applied after a predefined number of unimproved iterations (the number of tries fail to improve the current solution). As a result, the proposed algorithm shows good performance when compared with the original MFO algorithm and with state-of-the-art approaches.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Malek Alzaqebah,

Basic and Applied Scientific Research Center,

Imam Abdulrahman Bin Faisal University,

P.O. Box 1982, 31441, Dammam, Saudi Arabia

Email: maafehaid@iau.edu.sa

1. INTRODUCTION

Feature selection methods work by selecting the most relevant feature subset among a number of features in the dataset, which leads to better learning performance. Eliminating some features does not mean they are valueless information, but they may have considerable statistical relations with other features [1]. Feature selection methods are important during analysis and evaluation. Many feature selection algorithms have been discovered and widely used by scientists and researchers in experimental. Methods for feature selection are divided into three types depending on their relations with the classifiers [2, 3], these types are: The filter method works on overall characteristics of the data regardless of the classifier select the valuable features. The wrapper methods use optimization techniques to optimize the prediction process and the selected features. And the embedded methods, in the embedded method the feature selection is connected to the classification having the advantages of wrapper method which contain the interaction with the classification, while filter methods are less consumption of computer resources than wrapper methods [2-4]. Yet, this type is much robustness than in the wrapper method. That is because feature selection is included in the classifier architecture and the classifier is used to deliver a criterion for feature selection [5].

Various optimization algorithms have been employed as a wrapper-based method to solve the feature selection problem. Optimization algorithm considered as a technique that used to find the optimal solution for a certain problem [6]. However, the traditional optimization techniques faced a lot of challenges in solving recent complex problems. Therefore, metaheuristic algorithms as new optimization techniques courage the researchers to solve complicated problems in various fields by reducing the search time and the cost to find optimal solutions. Metaheuristic algorithms overcome the limitations in traditional optimization methods in three aspects; firstly, the metaheuristic algorithms do not require a gradient algorithm to find the solution of the optimization problems. Secondly, they are considering the Simplified method to apply in various fields that need to find the optimal solution. Lastly, they have a robust ability in exploration and reduce the probability of trapping in local optima [7].

A heuristic algorithm is considering a technique that can discover the fittest solution together can reduce the computational cost; however, this solution does not guarantee to be the optimal solution of the problem [8]. On the other hand, the metaheuristic algorithm is considering the enhancement of the heuristic algorithm that can make a mixture of random algorithms and local search algorithms. Fausto et al., define a meta-heuristic algorithm as a particular heuristic-based method [9]. The idea behind the heuristic-based method, it can solve various problems regardless of the basic algorithm framework. Metaheuristic methods can be divided into three main categories as stated by Khalilpourazari and Khalilpourazary in 2019, [10]: evolutionary algorithms, swarm intelligence, and physical-based algorithms.

Evolutionary algorithms are the group of algorithms that mimic the evolution behavior in nature, as an example of such algorithms are the Genetic Algorithm [11], Evolution Strategy, Genetic Programming and Biogeography-Based Optimizer [12]. Swarm-based algorithms are naturally inspired algorithms that mimic the group's animal behavior in searching for food, for example, The Grey Wolf Optimizer proposed by [13], Particle Swarm Optimization algorithm [14], Moth-Flame Optimization [15] and the Artificial Bee Colony algorithm [16]. Swarm-based algorithms have been widely used to solve optimization problems and showed to be superior in many fields of research [17-20].

Physical based algorithms utilize the physical rules in the real life for solving the optimization problems, examples are: The Simulated Annealing which simulates the annealing process of material. The law of gravity used by Gravitational Search Algorithm [21], The physical process of finding the damaged parts the structure is used by The Charged System Search [22], Curved Space Optimization [23] and the theory of multiverses inspired the development of Multiverse Optimization [24].

In 2015, Mirjalili adopted the moth-flame optimization algorithm (MOF) as a new swarm optimization algorithm [15]. MOF algorithm mimics the space exploration process of moths flying around the moonlight, MOF considered as an optimization technique developed based on the spiral flight of moths. It does not require the gradient information of the given problem. Nevertheless, the MOF algorithm has some limitations, for example, the low precision and sensitivity to a local optimum. Thus, it needs further investigation.

To overcome the limitations of the MOF algorithm the simulated annealing (SA) is hybridized with the MOF algorithm to enhance the global search capability of the MOF algorithm and to prevent getting trapped into local optimum. SA algorithm [25] gives the ability to skip out of local optima. Also, the SA algorithm has the benefit of accepting the worst solution with a specific probability, that can efficiently enhance other algorithms [26].

There are different modifications of the MFO algorithm to solve different problems. Moreover, there are various fields that can take benefit of this algorithm. In machine learning, gradient-based algorithms usually used in training and in general unable to solve these problems because they need gradient information of the problem and suffer from local optima. Li, et. al., 2016, use the MFO algorithm for tuning the parameters of the Support Vector Machine. Moreover [27], it was used during the training of the neural network to adjust the network's parameters, for example, the number of the hidden nodes, biases, and weights [28].

In the medical field, MFO has significant contributions especially in cancer detection [29, 30]. Another important proposed work was using MFO in feature selection [31]. On the other hand, several authors try to enhance the performance of the MFO algorithm by hybridizing it with Simulated Annealing, Particle Swarm Optimization, Gravitational Search Algorithm, and Firefly Algorithm. In this paper we proposed the Modified MFO algorithm with neighborhood search methods, which applied after a predefined number of unimproved iterations, as a result, the proposed algorithm shows good performance when compared with the original MFO algorithm and with of the state-of-the-art approaches.

2. PROPOSED ALGORITHMS

This section discusses the Moth-Flame Optimization algorithm and the proposed Modified MFO Algorithm for feature selection problem.

2.1. Moth-flame optimization algorithm

Moth Flame Optimization (MFO) is considered as a population-based metaheuristic algorithm developed by Mirjalili in 2015 [15]. MFO Algorithm mimics the behavior of Moths in nature during the transverse movement of the moths [15]. The real Moths use an intelligent flying method during the night to move in a straight line for long distances, by keeping a fixed angle toward the moon as a source of light. However, when the moths see an artificial light, they keep similar angle then they stuck in a spiral path around it because the artificial light is nearby compared to the light of the moon, such behavior is shown in Figure 1, where each moth eventually converges with respect to the light [15].

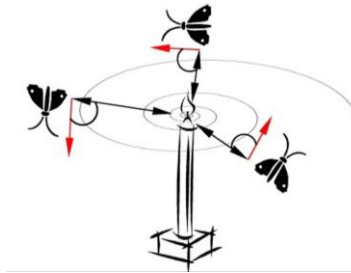


Figure 1. Spiral flying behavior around artificial light [15]

MFO algorithm consists of two main components the Moths and Flams, each Moth represents a candidate solution and the variables of a given problem are the position of moths in the space. The Moth is considered as a search agent that obtains the search. So, the moths are able to search in multiterminal space by updating their positions. A set of moths can be represented in the following array:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & \cdots & \cdots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,d} \end{bmatrix}$$

an array to keep the corresponding fitness value for each moth as follow:

$$OM = [OM1, OM2, OM3 \dots \dots OMn]$$

where n is the moths number and the dimension flames d is represented in an array to maintain the best position found so far by each moth, which is similar to the moth's Array as follows:

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & \cdots & F_{1,d} \\ F_{2,1} & F_{2,2} & \cdots & \cdots & F_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{n,1} & F_{n,2} & \cdots & \cdots & F_{n,d} \end{bmatrix}$$

Similar to moth's fitness array, an array to keep the corresponding fitness value for each best position found so far as follows: OF= [OF1, OF2, OF3 OFn]. By having both Moths and Flams arrays, each moth searches around and updates the flame (best positions) if a better solution found. The general representation of the MFO algorithm can be presented as follows: MOF= (I, P, T).

MOF algorithm has three tuple estimation procedures, I is a procedure that randomly initializes the population by the following formula:

$$M(i, j) = (ul(i) - ll(i)) \times rand() + ll(i) \quad (1)$$

In (1) the $ul(i)$ and $ll(i)$ are the upper and lower bounds of the variable I , and the objective function of moths is given by:

$$OM = \text{fitness function}(M); \quad (2)$$

Represents a procedure that responsible of searching for neighbor solutions of the moths until the T termination condition is met. Where T represents a procedure that returns whether termination condition is met or not. The main idea of the MFO algorithm is the model of transverse orientation behavior. The Moth updates its position in respect to a flame based on the following:

$$M_i = S(M_i, F_j)$$

where M_i is the moth with index i , and S indicates the spiral procedure and F_j is the flame with index j . For the MFO algorithm a logarithmic spiral is given by (3):

$$S(M_i, F_j) = D_i * e^{bt} * \cos(2\pi t) + F_j \quad (3)$$

where D_i represents the distance between the i th moth (M_i) and j th flame (F_j), calculated by: $D_i = |F_j - M_i|$, in (3), t is a number generated randomly and its value between -1 and b is the constant shape of the logarithmic spiral. The P procedure is the main procedure for the moths to explore the search space. The pseudocode of the MFO algorithm is shown in Figure 2.

```

//initialize the population of n solutions
for i = 1 to n
    for j = 1: d //d is the diminutions
        M(i, j) = (ul(i) - ll(i)) * rand() + ll(i)
    end
end
// MFO algorithm
while T(M) == false //Termination condition not met
    flame no=Random(N-1 * (N-1)/T)
    OM = FitnessFunction(M);
    if iteration == 1
        F = sort(M);
        OF = sort(OM);
    else
        F = sort(M_{t-1}, M_t);
        OF = sort(M_{t-1}, M_t);
    end
    for i = 1 to n
        for j = 1 to d
            Update r and t
            //t is a random number between -1 and 1.
            //r is the linearly decreased from -1 to -2
            D_i = |F_j - M_i|
            M_i = D_i * e^{bt} * cos(2\pi t) + F_j
        end
    end
end
end

```

Figure 2. The pseudo-code for MFO algorithm [15]

2.2. Modified MFO Algorithm for feature selection problem

MFO algorithm is used to select the optimal features from a given dataset. In MFO algorithm, after number of iterations, Moths (M) are influenced by their corresponding F (best position) and will stop moving towards F . If F is not changed, the Moths cluster around F . However, due to the solution of feature selection problem is an array consisting of 0 and 1, there is a great probability to generate the same or similar new individuals based on the updating strategy. If all or most individuals are replaced by the new generated individuals in each iteration, it will be difficult to keep the population diversity as the number of iterations increased and the algorithm may get trapped in local optima. To prevent Moths from getting trapped in local optima and to maintain population diversity, we propose simple three neighborhood methods as follows: NBChange, NBMove, and NBSwap.

In the Modified Moth algorithm we assumed the limit as the termination condition of the search if moths unable to improve the best solution after a number of iterations (limit), which means the Moths have gotten stuck in local optima. In order for Moths to leave the local optima after the parameter limit. We used the proposed neighborhood Methods to all Moths, then each Moth will start from its original position, and continue to search in the search space as shown in Figure 3.

```

//initialize the population of n solutions
for i = 1 to n
    for j = 1: d //d is the diminutions
        M(i, j) = (ul(i) - ll(i)) * rand() + ll(i)
    end
end
noImp=0
// Modified MFO algorithm
while T(M) == false //Termination condition not met
    flame no=Random(N-1 * (N-1)/T)
    OM = FitnessFunction(M);
    if iteration == 1
        F = sort(M);
        OF = sort(OM);
        bestF= OF[0]//save the best sol
    else
        F = sort(Mt-1, Mt);
        OF = sort(Mt-1, Mt);
        if OF[0] <= bestF //As maximization problem after //applying sort
            function the //best value will be the first
                noImp= noImp+1
            else
                bestF= OF[0]
        end
    end
    for i = 1 to n
        for j = 1 to d
            Update r and t
            Di =|Fj- Mi|
            Mi = Di * ebt * cos (2π t) + Fj
        end
    end
    if noImp >= limit then
        for i = 1 to n
            for j = 1: d
                M(i, j) = applyNeighborhoodMethods (M(i, j))
            end
        end
    end
end
end

```

Figure 3. The pseudo-code for the modified MFO algorithm

The solution of feature selection problem can be represented as a binary vector, for example, if the dataset contains 10 features the vector will be as follow: Sol= [1, 0, 1, 0, 1, 0, 1, 1, 0, 1], 1 if feature is selected and 0 if not selected. By considering this solution the proposed neighborhood methods are explained as follow:

- NBChange neighborhood Method selects a random feature and changes its value by Not operator, for example, we assume that the randomly selected feature is the third feature in Sol which its value is 1, the not operator will change it to 0. So, the new solution will be as follow:

$$\text{Sol}^* = [1, 0, 0, 0, 1, 0, 1, 1, 0, 1]$$

- NBMove neighborhood Method selects a random feature and moves its position to a new position, example, we assume that the randomly selected feature is the first feature in Sol which its value is 1, then we move it to a new random position (let's assume the fifth feature), the new solution will be as follow:

$$\text{Sol}^* = [0, 1, 0, 1, 1, 0, 1, 1, 0, 1]$$

- NBSwap neighborhood Method selects two random positions (features) and swaps their values, for example, we assume that the randomly selected two features are the third and sixth features in Sol. So, the new solution will be as follow:

$$\text{Sol}^* = [1, 0, 0, 0, 1, 1, 1, 1, 0, 1]$$

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

The investigation of the effectiveness and robustness of the proposed Modified MFO is presented in this section. Also, the Modified MFO is compared with other population-based algorithms i.e. Genetic algorithm (GA), Particle Swarm Optimization algorithm (PSO) and Firefly Algorithm (FFA) which tested on 8 datasets with diverse characteristics.

The parameters that used are the number of iteration where it is equal 100, the population size is 20 and the limit is 20. Table 1 shows a brief detail of the eight datasets used in this work. Which they are well known standard datasets retrieved from the UCI data source [32]. These datasets have been considered in several well- confirmed works. The main attributes for these datasets are the number of features (features), the number of instances (Instances) and the number of classes as shown in Table 1.

Table 1. List of the used datasets

Dataset name	features	Instances	Class
German	20	1000	2
Heart	13	270	2
Ionosphere	34	351	2
Parkinsons	23	197	2
SpectF	44	267	2
Sonar	60	208	2
WDBC	31	569	2
WBC	10	699	2

In this study, the instances in the datasets are split into training and testing, where 80% of the instances were applied for training purposes and 20% is used for testing purposes by Friedman et al., in 2001 [33]. All the runs and the experimental results in this research are prepared and reported on a PC with Intel CPU i5-5200U 2.2GHz and a RAM of 8.0 GB. The number of iterations used in this study was 100 and the population size equal to 10. Moreover, the average results obtained from 30 independent runs.

Table 2 reveals the number of selected features denoted as # of features and the best-obtained Accuracy denoted as ACC that are used to compare the modified MFO algorithm versus GA, PSO algorithm, FFA, and MFO algorithm. From Table 2 it can be seen that the Modified MFO algorithm can comparatively outperform other algorithms in terms of ACC with 50%, and comparable (same accuracy) with 50%. The GA couldn't achieve high accuracy compared with the PSO, FFA and MFO algorithms which obtain 3, 1, and 4 same ACC results obtained from the Modified MFO algorithm respectively.

Table 2 also shows that the Modified MFO algorithm outperforms other algorithms in term of the number of features just in 3 datasets, where the PSO achieve best results in 5 datasets. Figure 4 illustrates the comparison between GA, PSO, FFA, MFO and Modified MFO in terms of best accuracy and number of selected features. Table 3 displays the average Accuracy results that have been achieved by the GA, PSO, FFA, MFO and modified MFO algorithm. The Modified MFO algorithm obtained five best average results in German, SpectF, Sonar, Parkinsons, and WBC datasets, while in the heart dataset the MFO and the modified MFO algorithm achieve the same average, the PSO and MFO algorithms obtained the highest average in Ionosphere and Heart datasets respectively. The highest average accuracies are presented in bold.

Figure 4 (a and b) compare Modified MFO and other techniques depending on the best-obtained accuracy and the number of features selected. Figure 4 displays that the proposed technique was very close in accuracy results with comparable methods if not better in some cases such as Parkinsons, SpectF, Sonar and WBC databases. In the same sense, Figure 5 also compares between the proposed technique and the other approaches in term of the average accuracy of 30 runs for the best-obtained results and it was clear again that the results of the suggested method were equal to the best results or the best in accuracy in comparison with the other methods. Figure 4(b) shows that the proposed technique gave the best performance with most of the datasets in term of the reduction of the selected features

Table 2. Comparison of the best accuracy for the modified MFO algorithm with GA, PSO, FFA

Dataset	GA		PSO		FFA		MFO		Modified MFO	
	#features	ACC	#features	ACC	#features	ACC	#features	ACC	#features	ACC
German	11	78.00	13	78.38	8	77.88	12	78.63	13	78.63
Heart	6	88.03	5	88.42	5	88.42	5	88.42	5	88.42
Ionosphere	20	87.86	19	89.64	9	88.21	15	89.64	12	89.64
Parkinsons	14	88.42	10	89.00	7	88.50	10	88.42	7	89.71
SpectF	33	83.20	24	84.13	14	84.55	14	85.97	10	86.38
Sonar	36	84.52	33	86.28	17	83.90	12	86.32	17	88.09
WDBC	15	98.75	13	98.96	12	98.75	8	98.96	9	98.96
WBC	6	98.17	6	98.17	6	98.17	7	98.17	7	98.35

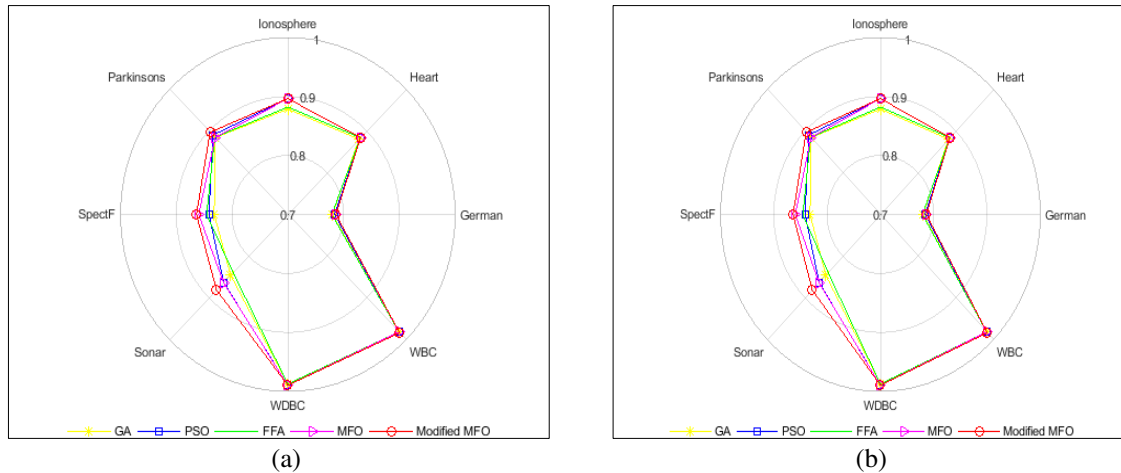


Figure 4. Comparison between GA, PSO, FFA, MFO and modified MFO in term of best accuracy, (a) accuracy, (b) selected features

Table 3. Comparison of the average of 30 runs for the best obtained results for all algorithms

Dataset	GA	PSO	FFA	MFO	Modified MFO
	ACC	ACC	ACC	ACC	ACC
German	77.26	77.88	77.03	77.66	78.09
Heart	84.81	87.29	87.33	88.42	88.42
Ionosphere	87.21	89.18	87.75	88.75	88.86
Parkinsons	86.94	87.20	87.16	87.46	87.93
SpectF	79.16	80.91	80.35	80.94	81.29
Sonar	82.03	84.64	82.81	85.26	85.50
WDBC	95.22	96.70	96.59	96.95	96.90
WBC	97.86	97.88	97.86	98.17	98.21

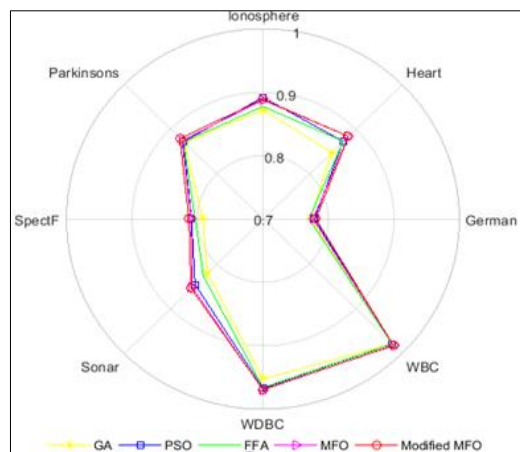


Figure 5. Comparison between GA, PSO, FFA, MFO and modified MFO in term of average accuracy of 30 runs

The Comparison for the Modified MFO algorithm with GA, PSO, FFA, and MFO algorithms by using tenfold cross-validation illustrated in Table 4, which reveals that the modified MFO algorithm achieve highest results in 3 out of 8 datasets (Ionosphere, Sonar, and WDBC datasets) and 3 datasets obtained same results in Modified MFO with one or more other algorithms (German, Parkinsons and WBC datasets). This finding indicates that 38% of the datasets obtain the highest accuracy in the Modified MFO algorithm compared with the gained results from other algorithms. While the FFA algorithm gains the highest accuracy in SpectF dataset and GA gain the best result in Heart dataset.

Table 4. Comparison for the ten-cross validation in the modified MFO algorithm with GA, PSO, FFA

Dataset	GA		PSO		FFA		MFO		Modified MFO	
	#features	ACC	#features	ACC	#features	ACC	#features	ACC	#features	ACC
German	11	81.00	14	82.50	20	81.50	18	81.50	14	82.50
Heart	9	96.29	7	92.59	7	92.59	6	90.74	7	92.59
Ionosphere	18	88.73	19	91.54	9	92.95	17	92.95	18	94.36
Parkinsons	17	91.02	6	92.30	7	90.38	3	92.30	7	92.30
SpectF	33	90.74	24	92.59	14	96.29	14	90.74	9	92.59
Sonar	43	83.33	33	83.33	20	80.95	20	88.09	12	88.09
WDBC	15	97.52	15	98.34	7	98.34	15	97.31	10	98.55
WBC	6	99.27	4	97.81	6	99.27	6	99.27	5	99.27

Figure 6 (a and b) compares the Modified MFO and other approaches using the ten-cross validation depending on the accuracy and the number of features selected respectively. From Figure 6 it was observed that the proposed technique is extremely acceptable in terms of accuracy and number of selected features. Table 5 shows the comparison of the results that have been obtained from the different approaches with those achieved from the Modified MFO algorithm based on the average accuracy results. The Modified MFO algorithm achieves five best average results in German, SpectF, Sonar, WDBC and WBC datasets. The highest average accuracies are presented in bold.

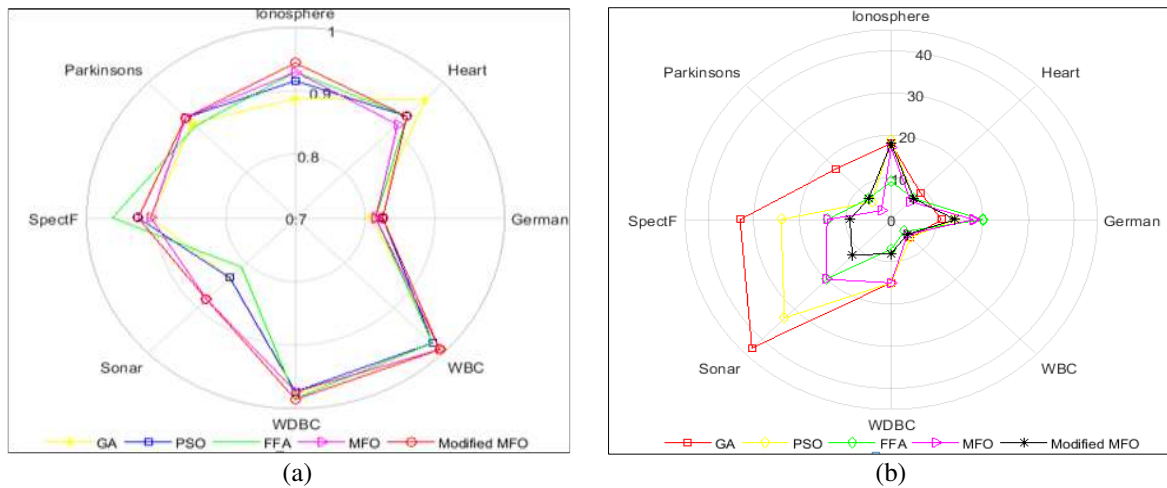


Figure 6. Comparison for the ten-cross validation in the modified MFO algorithm with GA, PSO, FFA in term of accuracy and selected features, (a) accuracy, (b) selected features

Table 5. Comparison of the average of 30 runs for the results using ten cross validation for all algorithm

Dataset	GA		PSO		FFA		MFO		Modified MFO	
	#features	ACC	#features	ACC	#features	ACC	#features	ACC	#features	ACC
Average	15.9	72.79	13.8	71.96	10.3	72.45	10	74.79	11.8	75.08
German	15.9	72.79	13.8	71.96	10.3	72.45	10	74.79	11.8	75.08
Heart	8.9	74.35	5.8	79.21	5.3	79.03	5.5	79.58	5.3	79.07
Ionosphere	22.6	86.06	16.8	87.89	10.9	87.89	14.9	90.99	17.1	90.14
Parkinsons	13.4	84.81	10.3	85.06	8	85.45	6.6	88.14	6.6	87.69
SpectF	31.6	68.24	21	66.25	10.2	74.86	10.4	83.24	13.9	84.26
Sonar	38.9	81.67	31.2	80.00	18	76.19	15	82.14	19.5	83.33
WDBC	17.3	93.20	14	95.10	9.7	95.04	7.6	97.07	6.5	97.09
WBC	6.8	95.78	6.2	95.42	6.5	95.58	6.4	95.77	5.4	96.35

Figure 7 shows the comparisons of the average of 30 run cross-validation for all algorithms in terms of accuracy and the number of features selected respectively. As observed from Figure 7 it was clear that the suggested algorithm had a good efficiency and made an acceptable improvement in reducing the number of features selected while maintaining a high accuracy ratio. To study the reliability and stability of the obtained results and to gain a sense of the solution accuracy, Boxplots of accuracy and number of selected features by the Modified MFO algorithm versus other algorithms on different datasets are compared and displayed in Figures 8 and 9. On each box, the red line indicates the median and the lowest and upper

borders of the box indicate the first and third percentiles, respectively. The whiskers expand to the farthest excessive data points not considered outliers, and the outliers are plotted separately using the '+' character. The distribution of the shape, diffusion and outside values are represented by the line extension.

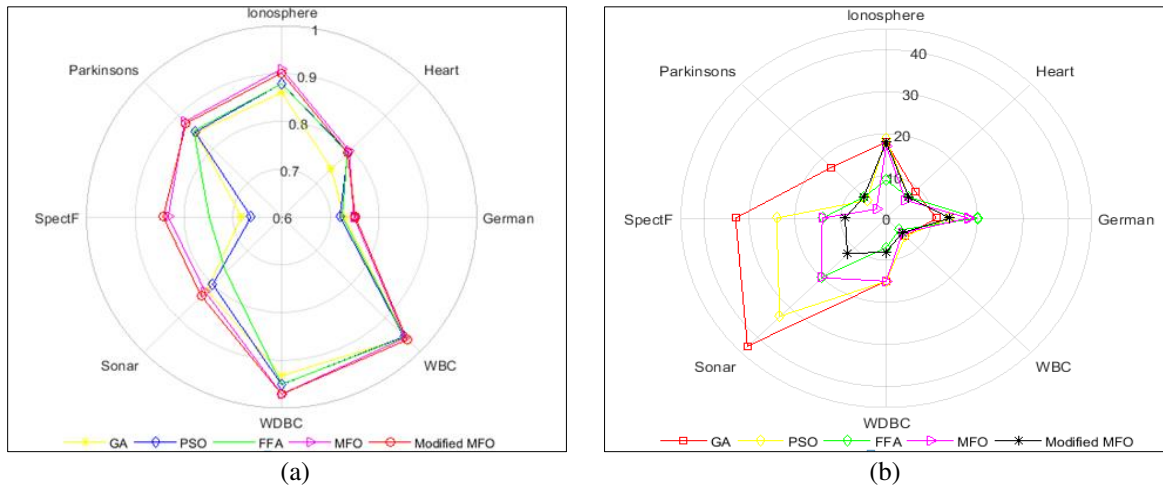


Figure 7. Comparison of the average of 30 runs for the results in ten cross validation for all algorithms in term of accuracy and selected features

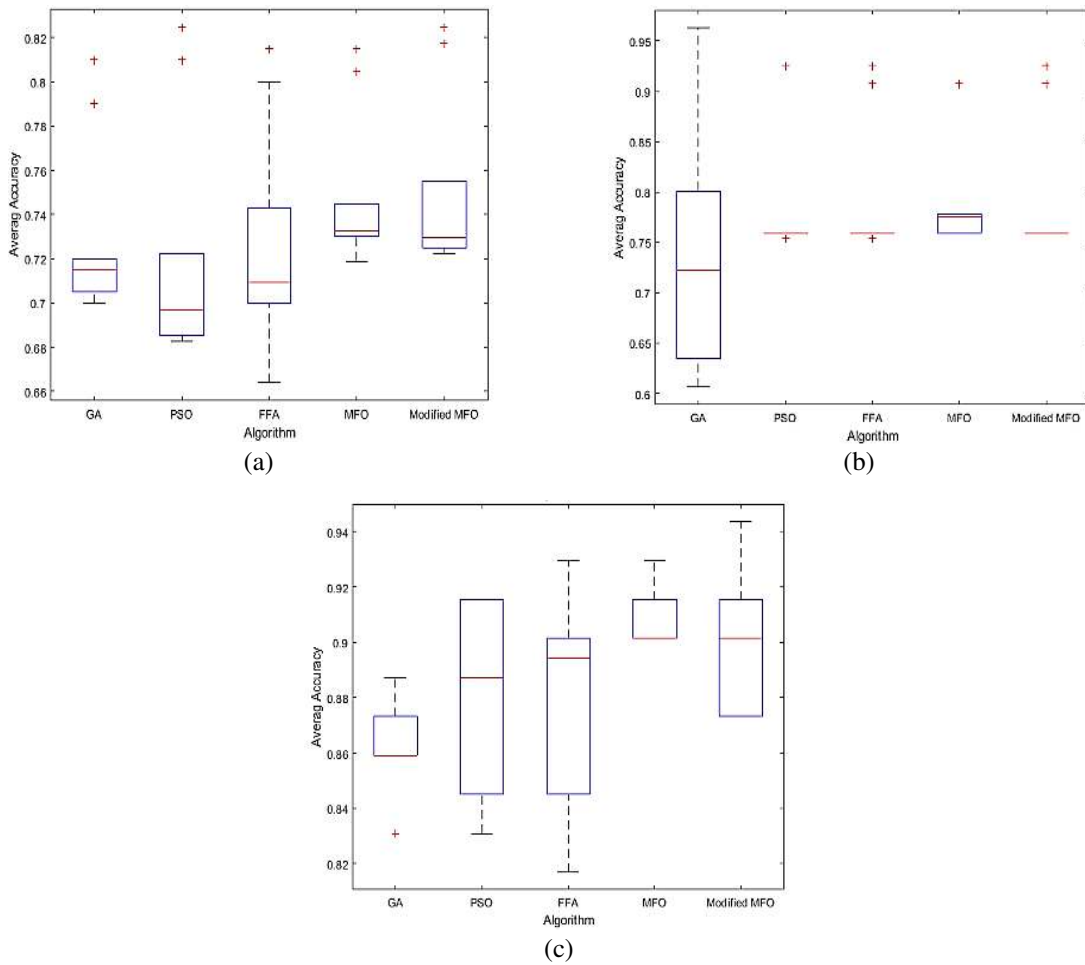


Figure 8. Boxplots of average accuracy results for Modified MFO versus other competitors applied to (a) German, (b) Hear and (c) Ionosphere datasets

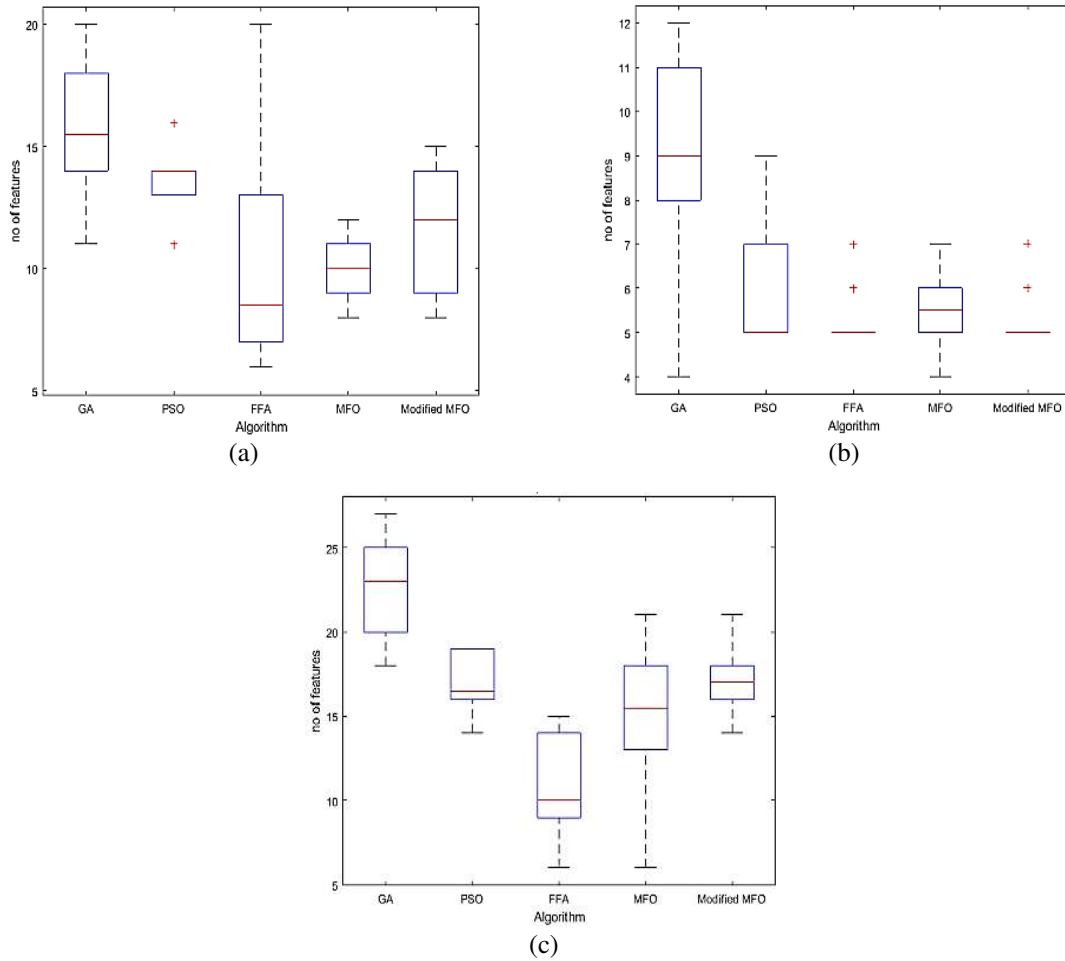


Figure 9. Boxplots for selected features for modified MFO versus other competitors on (a) German, (b) Hear, (c) Ionosphere datasets

To determine if the obtained results are significant or not, the Mann Whitney statistical test was applied by McKnight, and Najab in 2010 [34]. The p-values of the Mann Whitney statistical test based on suitability values are stated in Tables 6 and 7. Such statistical tests can prove that the spotted differences and improvements are significantly meaningful. Table 6 displays the excellence of Modified MFO in terms of average accuracy over the other competitors and it is statistically significant for most cases except for MOF technique and some cases in other approaches. The levels of marginal significance (p-values) of the Mann Whitney test according to the number of features are shown in Table 7. Where the observed differences between the Modified MFO and GA algorithms are statistically significant for all datasets and are statistically significant for most other competitor techniques except for the MOF technique. Figure 10 shows the convergence behavior of a. MOF and b. the Modified MOF, the x-axis denotes the number of iterations and the y-axis denotes the error rate measured by the SVM classifier.

Table 6. P-values of Mann Whitney test for the average accuracy results of modified MFO and other algorithms (p ≥ 0.05 are in bold)

Datasets	GA	PSO	FFA	MFO
German	0.009	0.015	0.035	0.796
Heart	0.075	0.023	0.015	0.280
Ionosphere	0.001	0.315	0.105	0.436
Parkinsons	0.315	0.796	0.089	0.739
SpectF	0.009	0.015	0.063	0.796
Sonar	0.089	0.007	0.000	0.165
WDBC	0.002	0.165	0.035	0.481
WBC	0.393	0.019	0.023	0.579

Table 7. P-values of Mann Whitney test for the selected features results of modified MFO and other algorithms (p ≥ 0.05 are in Bold)

Datasets	GA	PSO	FFA	MFO
German	0.007	0.143	0.218	0.143
Heart	0.002	0.631	1.000	0.481
Ionosphere	0.000	0.853	0.000	0.315
Parkinsons	0.000	0.002	0.105	0.912
SpectF	0.001	0.063	0.280	0.436
Sonar	0.000	0.000	0.853	0.105
WDBC	0.000	0.000	0.003	0.631
WBC	0.003	0.089	0.009	0.019

From Figure 10 its easily can be seen that the MFO algorithm has very fast convergence speed and it gets stuck in premature convergence [35]. Dorronsoro et, al.,2013 stated that when the algorithm performs a fast convergence, it's possible to get stuck in local optimal [36]. For example, first graph in Figure 10 is the behavior of MFO and Modified_MOF for the parkinsons dataset the solution not enhanced from iteration number 5 to 38. While in Modified_MOF for the same dataset, the algorithm smoothly converges and produces better results. In MFO algorithm at iteration# 5 the algorithm is able to produce error rate of 13.54 but in Modified MFO produces the same error rate after Iteration# 70, this shows that after applying the neighborhood methods to the solutions that unable to get better results, the converges speed of the solutions are slowing down a bit to search more effectively in search space and produces at the last better results. The Modified MFO technique showed equal to or better accuracy results and a better number of selected features for most of the used datasets but not all the observed differences have statistically significant in comparison with all other competitors.

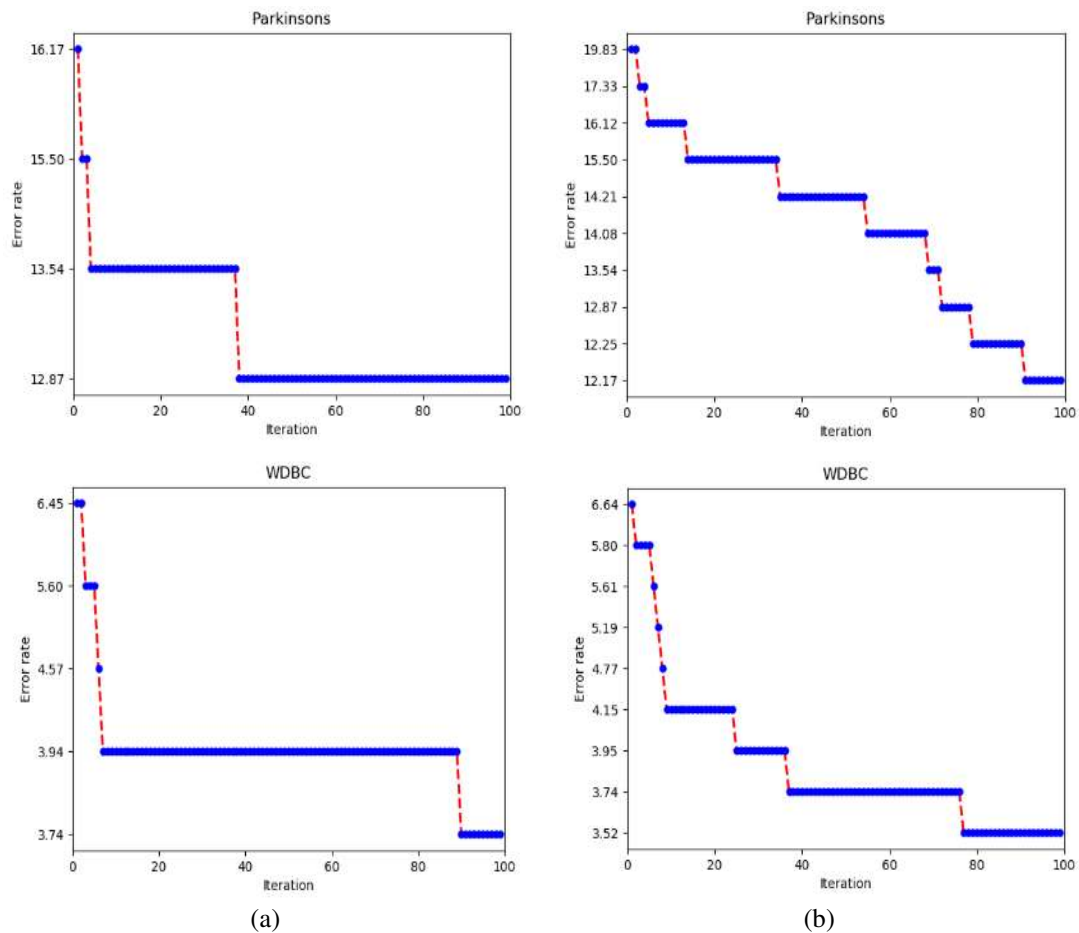


Figure 10. The convergence behavior of (a) MOF and (b) the modified_MOF

4. COMPARISON OF MODIFIED MFO ALGORITHM WITH THE STATE-OF-THE-ART APPROACHES

This section compares the best results obtained from the Modified MFO algorithm with the best-known solutions in the literature for the eight tested datasets. Table 8 compares the best-known results of the Modified MFO algorithm with those of the different algorithms from the literature. Accuracy is used as the main objective in comparing the performance of the algorithm. The best Accuracy is presented in bold. As observed from Table 8 the Modified MFO achieved very close values for most competitors in term of accuracy and better than others with some datasets Take into consideration the results taken from the different algorithm.

Table 8. Comparison between the modified MFO algorithm with state-of-the-art approaches

Dataset	Modified MFO	Best-known result	Source
German	78.63	78.00	Boughaci & Alkhalwaldeh 2018 [37]
Heart	88.42	83.30	Mafarja et al. 2018 [38]
Ionosphere	89.64	89.90	Mafarja et al. 2018 [38]
Parkinsons	89.71	92.00	Kumar & Kumar 2017 [39]
SpectF	86.38	82.60	Mafarja et al. 2018 [38]
Sonar	88.09	91.20	Mafarja et al. 2018 [38]
WDBC	98.96	97.00	Kumar & Kumar 2017 [39]
WBC	98.35	96.90	Aalaei et al. 2016 [40]

5. CONCLUSION

In this paper, the Modified MFO Algorithm with neighborhood search methods for feature selection problems. The algorithms in this work are applied on the benchmark of 8 standard UCI datasets were used. The results of the modified MFO algorithm were compared with four methods in the literature. This method demonstrated the superiority as a result by help in avoiding the premature convergence and help the algorithm jump-out from local optima, it was experiential that the neighborhood search methods suitable to improve the results in the proposed algorithm were it shows good performance when compared with the basic MFO algorithm and with the state of the arts approaches.

REFERENCES

- [1] Koturwar, P., Girase, S., and Mukhopadhyay, D., "A survey of classification techniques in the area of big data," *arXiv preprint arXiv:1503.07477*, 2015.
- [2] Saeys, Y., Inza, I., and Larrañaga, P., "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [3] Birkle, P., Zouch, M., Alzaqebah, M. and Alwohaibi, M., "Machine Learning-based Approach for Automated Identification of Produced Water Types from Conventional and Unconventional Reservoirs," *In Petroleum Geostatistics 2019, European Association of Geoscientists & Engineers*, vol. 2019, no. 1, pp. 1-5, Sep. 2019.
- [4] Bolón-Canedo, V., Sánchez-Marño, N., Alonso-Betanzos, A., Benítez, J.M., and Herrera, F., "A review of microarray datasets and applied feature selection methods," *Information Sciences*, vol. 282, pp. 111–135, 2014.
- [5] Shreem, S.S., Abdullah, S., Nazri, M.Z.A., and Alzaqebah, M., "Hybridizing ReliefF, MRMR filters and GA wrapper approaches for gene selection," *J. Theor. Appl. Inf. Technol.*, vol. 46, no. 2, pp. 1034-1039, 2012.
- [6] Khalilpourazari, S., and Khalilpourazary, S., "Optimization of production time in the multi-pass milling process via a Robust Grey Wolf Optimizer," *Neural Computing and Applications*, vol. 29, no. 12, pp. 1321–1336, 2016.
- [7] Mehne, S.H.H., and Mirjalili, S., "Moth-flame optimization algorithm: Theory, literature review, and application in optimal nonlinear feedback control design," *In Studies in Computational Intelligence, Springer Verlag.*, vol. 811, pp. 143–166, 2020.
- [8] Yang, X., Luo, Q., Zhang, J., Wu, X., and Zhou, Y., "Moth swarm algorithm for clustering analysis," *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, vol. 10363 LNAI, pp. 503–514, 2017.
- [9] Fausto, F., Reyna-Orta, A., Cuevas, E., Andrade, Á.G., and Perez-Cisneros, M., "From ants to whales: metaheuristics for all tastes," *Artificial Intelligence Review*, 2019.
- [10] Khalilpourazari, S., and Khalilpourazary, S., "An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems," *Soft Computing*, vol. 23, no. 5, pp. 1699–1722, 2019.
- [11] Holland, J.H., Langton, C., Wilson, S.W., Varela, F.J., Bourguine, P., and Koza, J.R., "Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)," *A Bradford Book, the MIT Press*, 1992.
- [12] Simon, D., "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [13] Mirjalili, S., Mirjalili, S.M., and Lewis, A., "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [14] Eberhart, R., and Kennedy, J., "A new optimizer using particle swarm theory," *In MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science IEEE*, pp. 39–43, 1995.
- [15] Mirjalili, S., "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [16] Karaboga, D., and Basturk, B., "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [17] Zhang, T., Yang, C., and Zhao, X., "Using Improved Brainstorm Optimization Algorithm for Hardware/Software Partitioning," *Applied Sciences*, vol. 9, no. 5, pp. 866, 2019.
- [18] Alzaqebah, M., Jawarneh, S., Sarim, H.M., and Abdullah, S., "Bees Algorithm for Vehicle Routing Problems with Time Windows," *International Journal of Machine Learning and Computing*, vol. 8, no. 3, pp. 234-240, 2018.

- [19] Alzaqebah, M., and Abdullah, S., "Hybrid bee colony optimization for examination timetabling problems," *Computers & Operations Research*, vol. 54, pp. 142-154, 2015.
- [20] Alzaqebah, M., Abdullah, S. and Jawarneh, S., "Modified artificial bee colony for the vehicle routing problems with time windows," *SpringerPlus*, vol. 5, no. 1, pp. 1298, 2016.
- [21] Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S., "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.
- [22] Kaveh, A., and Talatahari, S., "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267-289, 2010.
- [23] Moghaddam, F. F., Moghaddam, R. F., and Cheriet, M., "Curved Space Optimization: A Random Search based on General Relativity Theory," *arXiv:1208.2214*, 2012.
- [24] Mirjalili, S., Mirjalili, S. M., and Hatamlou, A., "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495-513, 2016.
- [25] Chen Hua-gen, Wu Jian-sheng, Wang Jia-lin, C. B., "Mechanism Study of Simulated Annealing Algorithm," 2004.
- [26] Zhang, Q., Liu, L., Li, C., and Jiang, F., "Moth-flame optimization algorithm based on adaptive weight and simulated annealing," *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Springer Verlag*, vol. 11266 LNCS, pp. 158-167, 2018.
- [27] Li, C., Li, S., and Liu, Y., "A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting," *Applied Intelligence*, vol. 45, no. 4, pp. 1166-1178, 2016.
- [28] Dosdoğru, A.T., Boru, A., Göçken, M., Özçalıcı, M., and Göçken, T., "Assessment of Hybrid Artificial Neural Networks and Metaheuristics for Stock Market Forecasting," *Sosyal Bilimler Enstitüsü Dergisi*, Cilt, vol. 24, 2018.
- [29] Sayed, G.I., and Hassanien, A.E., "Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images," *Applied Intelligence*, vol. 47, no. 2, pp. 397-408, 2017.
- [30] Sayed, G.I., Soliman, M., and Hassanien, A.E., "Bio-inspired swarm techniques for thermogram breast cancer detection," *In Studies in Computational Intelligence*, Springer Verlag, vol. 651, pp. 487-506, 2016.
- [31] Zawbaa, H.M., Emary, E., Parv, B., and Sharawi, M., "Feature selection approach based on moth-flame optimization algorithm," *In 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, Institute of Electrical and Electronics Engineers Inc., pp. 4612-4617, 2016.
- [32] Frank, A., and Asuncion, A., "UCI Machine Learning Repository," *Irvine, CA: University of California, School of Information and Computer Science*, 2010.
- [33] Friedman, J., Hastie, T., and Tibshirani, R., "The elements of statistical learning," New York: *Springer series in statistics*, vol. 1, no. 10, 2001.
- [34] McKnight, P.E. and Najab, J., "Mann-Whitney U Test," *In the Corsini Encyclopedia of Psychology* (eds I. B. Weiner and W. E. Craighead), 2010.
- [35] Jangir, P., "Optimal power flow using a hybrid particle Swarm optimizer with moth flame optimizer," *Global Journal of Research in Engineering*, 2017.
- [36] Dorronsoro, B., Burguillo, J.C., Peleteiro, A., and Bouvry, P., "Evolutionary algorithms based on game theory and cellular automata with coalitions," *In Handbook of Optimization*, Springer, Berlin, Heidelberg, pp. 481-503, 2013.
- [37] Boughaci, D. and Alkhawaldeh, A.A.S., "Three local search-based methods for feature selection in credit scoring," *Vietnam Journal of Computer Science*, vol. 5, no. 2, pp. 107-121, 2018.
- [38] Mafarja, M., Aljarah, I., Heidari, A.A., Hammouri, A.I., Faris, H., Ala'M, A.Z. and Mirjalili, S., "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Systems*, vol. 145, pp. 25-45, 2018.
- [39] Kumar, R.N. and Kumar, M.A., "A Novel Feature Selection Algorithm with Dempster Shafer Fusion Information for Medical Datasets," *International Journal of Applied Engineering Research*, vol. 12, no. 14, pp. 4205-4212, 2017.
- [40] Aalaei, S., Shahraki, H., Rowhanimanesh, A., and Eslami, S. "Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets," *Iranian journal of basic medical sciences*, vol. 19, no. 5, pp. 476, 2016.