



Solving feature selection problems by combining mutation and crossover operations with the monarch butterfly optimization algorithm

Mohammed Alweshah¹

Accepted: 25 September 2020 / Published online: 26 November 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Feature selection (FS) is used to solve hard optimization problems in artificial intelligence and data mining. In the FS process, some, rather than all of the features of a dataset are selected in order to both maximize classification accuracy and minimize the time required for computation. In this paper a FS wrapper method that uses K-nearest Neighbor (KSN) classification is subjected to two modifications using a current improvement algorithm, the Monarch Butterfly Optimization (MBO) algorithm. The first modification, named MBOICO, involves the utilization of an enhanced crossover operator to improve FS. The second, named MBOLF, integrates the Lévy flight distribution into the MBO to improve convergence speed. Experiments are carried out on 25 benchmark data sets using the original MBO, MBOICO and MBOLF. The results show that MBOICO is superior, so its performance is also compared against that of four metaheuristic algorithms (PSO, ALO, WOASAT, and GA). The results indicate that it has a high classification accuracy rate of 93% on average for all datasets and significantly reduces the selection size. Hence, the findings demonstrate that the MBOICO outperforms the other algorithms in terms of classification accuracy and number of features chosen (selection size).

Keywords Feature selection · Classification · Wrapper approach · Mutation operator · Crossover operator · Lévy flight. Monarch butterfly optimization

1 Introduction

The widespread application of information technology at all levels has contributed to an exponential growth in the volume of stored data [1]. These data sets contain a wealth of information that is not readily accessible, but which, if retrieved, may be of great value and significance. The extraction of this information from the data volume is called data mining [2–5]. The feature selection (FS) process is one of the important stages in data mining as it is used to reach this information in the fastest and most accurate way. The process involves selecting the features that best fit the study problem. So, instead of selecting all of the features, only the specific features related to the study problem are chosen [6]. This means that only those features that are important for making an informed

decision [7–9] about a particular problem are selected for further processing. This approach therefore allows for considerable classification accuracy [10] and also helps to improve the understanding of statistical structures and performance [11] in a shorter time.

The FS process is composed of three key steps [12]. First, the generation process generates subsets of the entire data array. This is followed by the assessment phase and the validation phase [13]. In FS, the features associated with a problem are selected by using either a filter or a wrapper approach during the assessment phase [14]. A number of specific approaches and selection criteria have been developed for the filter approach, such as mutual information [15], information gain [16], principal component analysis and others [17]. On the other hand, a classifier in wrapper form, such as the K-nearest Neighbor (KNN) and the support vector machine (SVM) [18], can also be utilized.

If, in the generation process all the possible subsets of a data set are obtained, this results in a problem with very high complexity that requires a high computation time of 2^n , where n is the number of features in a data set [19]. Many artificial intelligence (AI)-based methods have been used to

✉ Mohammed Alweshah
weshah@bau.edu.jo

¹ Prince Abdullah Bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt, Jordan

reduce the search time and at the same time ensure reliable and accurate results. One of the AI methods that is most widely used at the current time is the metaheuristic algorithm. This type of algorithm can be employed as a tool to search for patterns and traits so as to improve the performance of a classifier [20]. A metaheuristic algorithm can perform a random search, thus accelerating the speed of finding relevant features. Furthermore, to ensure accuracy, it takes into account the information gathered during the search to guide the ongoing search process [21]. There are several types of metaheuristic algorithm that are inspired by nature and have been proved to be effective [22, 23], in finding solutions for many AI problems [24, 25].

Metaheuristic algorithms are widely used in problem solving of FS [26–28]. They randomly generate subsets based on parameters that decide how those algorithms work [29], and the execution times have been shown to be dramatically reduced and consistent results provided [30, 31]. Some of the metaheuristics that have been used to solve FS problems include the mine blast algorithm (MBA), chaotic dragonfly algorithm [32], ant colony optimization [33], particle swarm optimization [34], bee colony optimization [35], grasshopper optimization [36], the chaotic crow search algorithm [37], evolutionary gravitational search algorithm [38], cat swarm optimization algorithm [39], harmony search algorithm [40] and cuckoo search algorithm [41] among others [8, 25, 31, 42–44].

Monarch butterfly optimization (MBO) is another metaheuristic algorithm that is nature inspired. The MBO is based on the migratory behavior of a North American butterfly, and was initially developed by Wang et al. [45]. The MBO has obtained satisfactory results when applied to many optimization problems and FS problems [46].

Building on this success, in this paper, the MBO is modified in two ways in order to try to achieve more reliable results. The first modification involves the use of an enhanced crossover operator, and is named MBOICO. The second modification involves the use of Lévy flight, and is named MBOLF. The performance of these two proposed methods is assessed and compared with four recent wrapper FS methods, namely, the original MBO, the hybrid whale optimization algorithm with SA (WOASAT), the mine blast algorithm with simulated annealing (MBA-SA), gravitational search algorithm (HGSA) and the (BGSA). The findings demonstrate that MBOICO is superior to these approaches in terms of classification accuracy and the number of features chosen (selection size).

The remainder of this paper is structured as follows: Section 2 provides an overview of the works that are the most relevant to this study. Next, section 3 describes the MBO algorithm. Then, section 4 introduces the proposed modified MBO methods, namely, MBOICO and MBOLF. This is followed by section 5, which describes the experiments and

discusses the results of those experiments. Finally, section 6 draws some conclusions and makes some suggestions for possible future research directions.

2 Related works

In the context of nature-inspired metaheuristics, the first noteworthy study is that by Wang et al. [47], who implemented the (EWA) algorithm which was inspired by the earthworm's two forms of reproduction (named Reproduction 1 and Reproduction 2). Reproduction 1 produces only one offspring, while Reproduction 2 generates one or more offspring at a time, and nine enhanced crossover operators can effectively do this. The researchers also added Cauchy mutation (CM) the EWA method. Nine different EWA methods based on nine improved crossover operators were implemented with one, two and three offspring, respectively, giving a total of 27 variations of the method. The findings indicated that EWA23 improved performance, and on most of the tested benchmark data sets, it was able to achieve a decent level of fitness as compared to the others. In another approach, Sayed et al. [48] introduced a hybridized model based on the slap swarm algorithm (SSA) and chaos theory. The researchers also evaluated the efficacy of tenchaotic maps as a means of improving the results of the basic SSA. The outcomes demonstrated that the SSA disorderly model greatly improved SSA efficiency in respect of both the global and local search.

Chatterjee et al. [49] implemented S-shaped and V-shaped conversion functions to transform the (SSD)'s continuous search space to binary form and also implemented a local search algorithm, named (LAHC), to improve the SSD's exploitation functionality. The results showed that this approach achieved high classification accuracy with low selection size. On the other hand, Ghosh et al. [50] instituted a mimetic algorithm for handwritten word recognition based on wrapper-filter FS. While, in another recent work, Shunmugapriya et al. [35] introduced a new hybrid ant and bee colony model for enhancement the FS problems. Zawbaa et al. [51], used (ALO) in a wrapper approach to optimize FS, whereas Shivalingegowda and Jayasree [52] proposed an approach to enhance the FS process in a wireless network by using the gravitational search algorithm.

Abdelbaset et al. [53] tried to solve the feature selection problems using mutation operator with the gray wolf algorithm. The findings demonstrate how the operator of the mutation increases the functionality and feasibility of this approach to enhance the feature selection process. Moreover, in [54] Wu implemented the Pareto optimization with ABC algorithm to pick non-dominant features from big data. On the other hand, Emary and Grosan [55] enhanced ALO performance for FS with use of chaos. To show its efficacy in reducing features of complex data, they adapted their approach

to biology and medicine data. In addition, Wang et al. [56] used data preprocessing with FS, using differential evolution (DE), and Shahbeig et al. [57] used binary DE for determining sponge iron consistency.

Mafarja et al. [36] implemented grasshopper optimization algorithm as well as evolutionary population dynamics in wrapper model the findings of which demonstrated the EPD's potential to greatly improve the GOA in terms of accuracy and selection size.

Another mechanism in FS, which was developed by Malakar et al. [58] implemented a GA-based hierarchical FS method for Indian handwriting recognition, the findings of which indicated a significant increase in accuracy of the letter recognizing. In another work, Saidi et al. [59] developed a hybrid solution that used the GA and the Pearson coefficient of correlation to improve the FS procedure; and Emary et al. [60] suggested a wrapper FS structure using a binary ant lion approach and demonstrated that a fast convergence speed along with high classification efficiency could be achieved through their system.

Several approaches were hybridized to address the problems with FS. For example, in [61, 62] a hybridization was introduced between the ACO and the GA. In addition, Mafarja and Abdullah [63] employed SA to enhance the GA local search. Also, Zorarpaci and Ozel [64] proposed a wrapper model hybrid of ABC algorithm and DE. On the other hand,

Zhang et al. [65] performed the first analysis of a cost-based function selection problem on multi-objective (PSOs) by integrating the hybrid operator, the crowding distance, and the Pareto dominance relationship. The researchers named their method HMPSOFS. The proposed multi-objective algorithm is matched on that HMPSOFS can more efficiently scan the solution space as a multi-objective algorithm in order to achieve multiple test data sets with three multi-objective FS algorithms. The experimental findings demonstrated a set of non-dominated solutions rather than a single one. Examining the Pareto front obtained by the multi-objective algorithm will help users select their favorite solutions to meet their own needs. So, it is a highly efficient way of selecting features to address cost-based selection problems.

Yong et al. [66] proposed a multi-objective selection algorithm based on bare-bones particle swarm optimization. Their approach relied on a fortified memory technique that was intended to solve the particle depletion phenomenon. That is, a hybrid mutant was used with aim of enhancing the search capabilities of the proposed algorithm.

On the other hand, Mafarja and Mirjalili [67] proposed a new hybrid integrating WOA and SA in Wrapper FS (WOASAT) process. SA was used in their research in two manners: during the first iteration, SA and WOA collaborated with each other to pick features as one portion. The SA process was used in the second step following the end of the WOA. The findings revealed that the WOASAT approach

improved the FS mechanism compared with new state-of-the-art methods in term of accuracy and selection size. In other research, Abualigah et al. [38] employed PSO as an FS approach, where a subset of features was extracted by using a clustering technique (k-means) to guarantee the clustering precision of the FS approach. This method strengthened the performance of textual record clustering because it used text clustering in k-means to identify further similar classes. The methodology was applied to six clustering data sets of textual data, and the findings were contrasted to the produced by certain excellently-known previous FS strategies. The performance evaluation showed that the new approach provided more reliable findings in terms of both classification accuracy and selection size.

In addition, Mafarja et al. [68] implemented the GOA to address FS problems with the use of two strategies based on transfer functions and the location of the better solution which keeps the current solution simple. The efficacy of the approaches introduced was contrasted with that of 11 existing methods using 25 comparison datasets. The findings revealed that the methods proposed improved the level of convergence in finding the best solutions from the features chosen relative to the methods before. Therefore, the mutation operators significantly improved the GOA.

In a recent study, Faris et al. [69] developed a crossover operator-driven scheme for solving FS problems that involved using improved binary SSA-based optimization. The researchers compared the performance of their proposed scheme with that of prior FS methods on 22 well-known data sets. The results showed that the proposed scheme had better detection precision than other methods such as BGWO, BBA and BPSO. Their approach also demonstrated the obvious dominance that convergence approaches had about hybrid methods.

Recently, Taradeh et al. [38] proposed a new approach using evolutionary gravitational search with two classifiers, KNN and DT, to solve FS problems. The results demonstrated that the algorithm had the ability to achieve high affinity rates through a balance between local and global search as compared to previous methods. The suggested HGSA can effectively outperform other wrapping approaches and demonstrate benefits in terms of exploration and exploitation, tradeoffs in search patterns and higher convergence speeds relative to other peers on multiple FS tasks.

Alweshah et al. [70] proposed two methods to enhance the FS process. The first involved the use of the MBA in order to automate the discovery step of the FS process. The second involved hybridizing the MBA with simulated annealing (MBA-SA) as a local search to enhance the MBA's ability to locate solutions in the exploitation phase. The performance of the two suggested approaches (MBA and MBA-SA) was evaluated on 18 benchmark data sets from the UCI repository, and the detailed experimental findings indicated that MBA-

SA obtained good results compared with other approaches in the literature. Later, Alweshah et al. [46] proposed a wrapper FS approach using the MBO. The experiments that were carried out showed that the MBO achieved a high classification accuracy rate and was able to dramatically reduce the number of selected features.

There are several other works which have employed MBO in varying fields and areas. Including Feng et al. [71] applied chaos theory and Gaussian mutation worker to boost MBO 's global search capabilities. Their suggested approach, called CMBO, was used as a binary optimization problem to solve three large-scale group instances of 0–1 knapsack problem. The findings revealed that their approach was capable of outperforming the simple MBO and eight other algorithms in determining the best outcome to this type of problem. Chakrabarty et al. [72] used MBO with images collected by satellites to identify information in the captured images, such as forest areas and different terrains. The findings revealed that their approach obtained a high similarity between the break and the initial images and therefore had a slightly improved ability to segment very complicated images compared to other methods of segmentation. In addition, In the meanwhile, The ABC algorithm and MBO algorithm were applied by Ghanemet al [73] to optimize the solution to numerical optimization problems by optimizing the equilibrium between local search and global search phases in order to achieve effective results in terms of precision, convergence speed and preventing local optimization. Their suggested hybrid approach, called ABC-MBO (HAM), was applied to 13 datasets of benchmarks and the results were equated with nine metaheuristic algorithms used to solve numerical optimization problems. The findings showed that the ABC-MBO (HAM) had the potential to obtain more reliable results and improve the level of convergence relative to the other processes.

In another domain, Devikanniga and Raj [74] used the MBO algorithm to refine the weights of an artificial neural network classifier (ANN), MBO was used to create a model to differentiate a stable person from an adult with osteoporosis. Experimental findings revealed that the new model had a high potential for interpretation of medical data compared to other related models of ANN hybridization.

Another early MBO study was undertaken by Strumberger et al. [75], who suggested an MBO-based hybrid strategy with FA. The approach was tested on six sample datasets, and the study proved that the new hybrid approach was more effective in solving hard optimization problems than the five methods it was compared to. Also, Faris et al. [76] employed MBO with combat premature convergence more effectively. To this end, they used a novel operator to extend the outcomes of the hunt according to the precision and speed of the execution. In addition they used their approach to teach expectations of multilayers. They implemented their approach in 15 benchmark datasets, and the results revealed an increase both in terms of

escaping local optima as well as a slightly faster convergence rate and shorten execution times.

Most of the preceding works used metaheuristic algorithms as a search approach for FS problems. Many of these works used these algorithms without modification. However, others modified them in several ways, and the most important of which was hybridization to improve the balance between local search and global search to prevent exposure to local optima. The MBO algorithm has been shown to obtain good results in terms of classification accuracy when it is used mostly without modification and this reflects the fact that there is a balance between global and local search in this algorithm. Nevertheless, in this paper, the global search attribute of the MBO algorithm was modified in order to achieve more precise results and to improve its efficiency in finding the correct points from the start and before turning to local search. This proposed methodology therefore differs from other approaches in the literature because with this method, the researcher aims to show the significance of global search in solving FS problems.

2.1 Monarch butterfly optimization

As mentioned above, MBO, which developed by Wang et al. [45], is one of the metaheuristic algorithms that mimics the behavior of fauna in the natural world. Specifically, it simulates the normal migration pattern of monarch butterflies that live in North America. Like many other butterflies, the monarch butterfly migrates twice a year. In the first migration, which starts in Canada the butterflies head south to Mexico, and in the second migration, the butterflies make the return journey from Mexico north to Canada [77].

The migratory activity of these butterflies is replicated to solve multiple optimization problems. However, certain rules and fundamental concepts must be followed so that the best solution to the problem can be found [78]. First, most of the butterflies that represent the population are either located in Land 1 (the pre-migration home) or in Land 2 (the post-migration home). Second, every butterfly's offspring is produced individually through the migration operator, regardless of whether the siblings are present in Land 1 or Land 2. Note also that the population does not change; it must remain stable. Therefore, at each iteration, a fitness function is used to eliminate either the parent or the new child. Lastly, the 54 selected butterflies based on the fitness feature are transferred towards the next generation and are not changed by the migration operator.

Around the beginning of April, butterflies start to migrate when they leave Land 1 and go to Land 2, and then, in September, the reverse migration process from Land 2 to Land 1 begins. The cumulative number of monarch butterflies in the two lands represents the entire population, which is called NP.

The individual butterflies interpret the data and communicate with each other locally, spreading information within the swarm, which results in the evolving capability of the system. Examining mental systems in this way is important for understanding intelligence. In such systems. This general viewpoint also makes one analyze swarms differently naturally inspired rather than depending on a thorough knowledge of the micro-level.

2.1.1 Migration operator

The Butterfly migration mechanism is described as follows [79]:

$$X_{i,k}^{t+1} = X_{r1,k}^t \quad (1)$$

where $X_{i,k}^{t+1}$ defines the Kth elements of X_i at generation $t + 1$,

describing the location of the butterfly land $X_{r1,k}^t$ denotes Kth elements from new generation place. In this, r is a random number which is defined by the given formulas:

$$R = \text{rand} * \text{peri} \quad (2)$$

Where, peri is the time span for migration.

In another case, if $r > p$, and Kth elements of the location of the new generation are then calculated by the equation:

$$X_{i,k}^{t+1} = X_{r2,k}^t \quad (3)$$

Where $X_{r2,k}^t$ defines X_{r2} Kth elements in t butterfly generation $r2$. And P indicates the ratio of monarch to land butterflies 1. An overview of a butterfly's migration operator is given in Algorithm 1.

Algorithm One : The Pseudocode of MBO Migration operator

```

1- Begin:
2-   for i = 1 to NP1 (for each butterfly in Sub-population no.1) do
3-     for k=1 to D (for each element in i th monarch butterfly) do
4-       generate a number of rand uniform distribution;
5-       r =rand*peri;
6-       if r ≤ p then select a monarch butterfly in Sub-population 1 randomly
(say r1);
7-         Generate the Kth element as Eq. (1).
8-         Else.
9-         select a monarch butterfly in Sub-population No.2 Randomly (say r2);
10-        Generate the kth element as Eq. (3).
11-        end if.
12-      end for k.
13-    end for i .
14- end.
```

2.1.2 Butterfly operator adjustment

Using this algorithm the equilibrium is achieved by evaluating the P value ratio between the migration directions from land 1 to land 2. If the value of P is big, it indicates that the amount of selected butterflies from Land 1 is higher than that from Land 2, and conversely.

The position of the butterflies is modified if the rand produced is less than or equal to the P value. The following equation indicates the Butterfly location's modified position:

$$X_{j,k}^{t+1} = X_{best,k}^t \quad (4)$$

where $X_{j,k}^{t+1}$ represents the Kth elements of X_j at $t + 1$ generation, that explains butterfly position j , and $X_{best,k}^t$ define the Kth elements of X_{best} in both Land 1 and Land 2 at present generation t . Thus, if $\text{rand} > P$, then the following formula shifts it:

$$X_{j,k}^{t+1} = X_{r3,k}^t \quad (5)$$

Then, when the rand is greater than BAR , the following equation will change the current location:

$$X_{j,k}^{t+1} = X_{j,k}^{t+1} + \alpha * (dx_k - 0.5) \quad (6)$$

Where the BAR reflects the change factor for the butterfly and the dx is the walking step for the j butterfly which is determined by flying the Lévy as follows:

$$Dx = Lévy \left(X_j^t \right) \quad (7)$$

And α in Eq. (6) is a weighted variable measured with Eq. 8:

$$\alpha = S_{max} / t^2 \quad (8)$$

Where S_{max} Represents the maximum walking butterfly length in a single step, and t represents the current generation. The butterfly-adjusting system definition remains set out in Algorithm 2.

Algorithm Two: The Pseudocode adjusting operator of MBO

- 1- Begin
 - 2- for j= 1 to NP2 (for each butterfly in Subpopulation NO. 2) do:
 - 3- Get the gait step dx using Eq. (7).
 - 4- Get (weighting factor) using Eq. (8).
 - 5- for k=1 to D (for each element in jth monarch butterfly): do
 - 6- make numbers by randomly using uniform distribution;
 - 7- if rand \leq p then get the kth element of the t 1 j x + as in the Eq. (4).
 - 8- else select the MBO into sub-population No.2 randomly (say r3);
 - 9- Generate the kth of element as in the Eq. (5).
 - 10- if rand > (BAR) then
 - 11- $X_{j,k}^{t+1} = X_{j,k}^{t+1} + \alpha * (dx_k - 0.5)$
 - 12- endif.
 - 13- end if.
 - 14- end for k.
 - 15- end for j.
 - 16- End.
-

The general behavior of the MBO algorithm is defined in algorithm 3 after studying the behavior of the butterflies in algorithms 1 and 2. [80].

Algorithm Three: The Pseudocode for the MBO algorithm

- 1- Step 2: While (G < Max Gen)
 - 2- Sort the population depending of fitness.
 - 3- Produce sub-population1 with NP1 better individuals and concept
 - 4- sub-population2 on the remainders.
 - 5- Generate the offspring solution for sub-population1.
 - 6- Generate the offspring solution for sub-population2
 - 7- Recombine the newly generated sub-populations into one population.
 - 8- Find the current best individual.
 - 9- End while.
 - 10- Step 3: Find the global-best individual.
 - 11- End.
-

3 Proposed approaches

In this research, two methods proposed used the wrapper FS technique. First one is using an optimized cross over operator (MBOICO) to strengthen the search capability of MBO. Second, the Lévy flight is using in MBO search.

3.1 MBO with improved crossover operator (MBOICO)

In the previously studied MBO algorithm, all the butterflies generated will move automatically to the next generation irrespective of their fitness value [81], that the predicted outcomes will reduce if the fitness interest is religious. In order to allow butterflies with better fitness values to only cross the next stage, the MBO Search was updated using a cross-operator by updating the migration operator and updating the butterfly modification operator in this paper.

3.1.1 Updating migration operator

To ensure that only butterflies with better fitness values arrive for optimum precision in the next generation, the first generation migration coefficient modification is used according to the formula below. [82]:

$$X_{i,new}^{t+1} = \begin{cases} X_{i}^{t+1}, & f(X_{i}^{t+1}) < f(X_{i}^t) \\ X_{i}^t, & \text{otherwise} \end{cases} \quad (10)$$

Where the $X_{i,new}^{t+1}$ is the update of the first generation. $f(X_{i}^{t+1})$ and $f(X_{i}^t)$ are the fitness values for butterflies X_{i}^{t+1} and X_{i}^t , respectively.

Algorithm 4: Update the Migration operator:

- 1- Begin
 - 2- for $i = 1$ to NP1 (for all of the monarch butterflies in Sub-population 1) do
 - 3- for $k=1$ to D (all basics in i th monarch butterfly) do
 - 4- Arbitrarily produce quantity rand by unchanging spreading;
 - 5- $r = \text{rand} * \text{peri}$;
 - 6- if $r \leq p$
 - 7- at that time randomly choice a monarch butterfly in Sub-population 1 (say r1);
 - 8- Produce the Kth component of the $X_{i,k}^{t+1}$ by Eq. 1.
 - 9- Else
 - 10- Randomly choice a monarch butterfly in Sub-population 2 (say r2);
 - 11- Produce the kth component of the $X_{i,k}^{t+1}$ Eq. 3.
 - 12- end if;
 - 13- end for k ;
 - 14- Produce $X_{i,new}^{t+1}$ based on the greedy strategy as in the Eq. 9.
 - 15- end for component i ;
 - 16- end;
-

3.1.2 Updating butterfly adjusting operator

In the optimization process, the self-adaptive crossover operator with greedy strategy is used to gain the maximum value of first-generation knowledge about the butterfly population. In the butterfly adjusting process, a new, powerful form of crossover operator developed. The updated version will formulate as equation as follows:

$$X_{j2}^{t+1} = X_{j1}^{t+1} \times (1 - \text{Cr}) + X_j^t \times \text{Cr} \quad (11)$$

Where X_{j2}^{t+1} is additional afresh-generated butterfly individual using X_{j1}^{t+1} and X_j^t . And Cr represents the crossover rate, and a self-adaptive process is used to change it as the following equation:

$$\text{Cr} = 0.8 + 0.2 \times \frac{f(X_j^t) - f(X_{best})}{f(X_{worst}) - f(X_{best})} \quad (12)$$

Where $f(X_j^t)$ is fitness function of butterfly j in Sub-population 2, and X_{best} and (X_{worst}) Reflect the worst and best individual butterfly in the entire butterfly population

respectively. And as shown Cr 's scale value varies from 0.20 to 0.80.

Now you can show the new butterfly as in the following:

$$X_{j,new}^{t+1} = \left\{ \begin{array}{ll} X_{j1}^{t+1} & , \quad f(X_{j1}^{t+1}) < f(X_{j2}^{t+1}) \\ X_{j2}^{t+1} & , \quad f(X_{j2}^{t+1}) < f(X_{j1}^{t+1}) \end{array} \right\} \quad (13)$$

Where $f(X_{j1}^{t+1})$ and $f(X_{j2}^{t+1})$ represents the fitness function of the butterfly X_{j1}^{t+1} and X_{j2}^{t+1} respectively. Algorithm 5 shows the updating butterfly adjusting operator.

Algorithm 5: Updating the Adjusting Butterfly Operator

```

1- Begin
2- for j= 1 to NP2 (for all in the monarch butterflies in Sub-population 2) do
3- Estimate the walk step operation dx by Eq. (7);
4- Estimate the factor of the weighting by Eq. (8);
5- for k=1 to D (for all of the elements in (jth) monarch butterfly) do
6- Generate randomly (rand number) using the uniform distribution;
7- if rand ≤ p then
8- Produce the kth element as in the Eq. (4).
9- Else
10- Randomly select a monarch butterfly in Subpopulation 2 (say r3);
11-Generate the kth element as Eq. (5).
12-if rand > BAR then
13- $X_{j,k}^{t+1} = X_{j,k}^{t+1} + \alpha * (dx_k - 0.5)$ 
14- endif
15- end if
16- end for k
17-Generate  $X_{j2}^{t+1}$  by execution SAR operator as shown in Eq. (10).
18-Generate  $X_{j,new}^{t+1}$  rendering to greedy strategy as in the Eq. (12).
19- end for j
20- End

```

The wrapper FS algorithm proposed in this paper uses a KNN classifier as an evaluator, while at the same time it uses MBOICO as a search technique. The KNN classifier specifies the precision limit for the features selected by MBOICO. Fig. 1 illustrates the steps involved in the MBOICO technique when using wrapper-based FS.

3.2 MBO with Lévy flight method (MBOLF)

In this study, another method was suggested to change the MBO search by developing a levey-flight mutation operator dependent mutation operator.

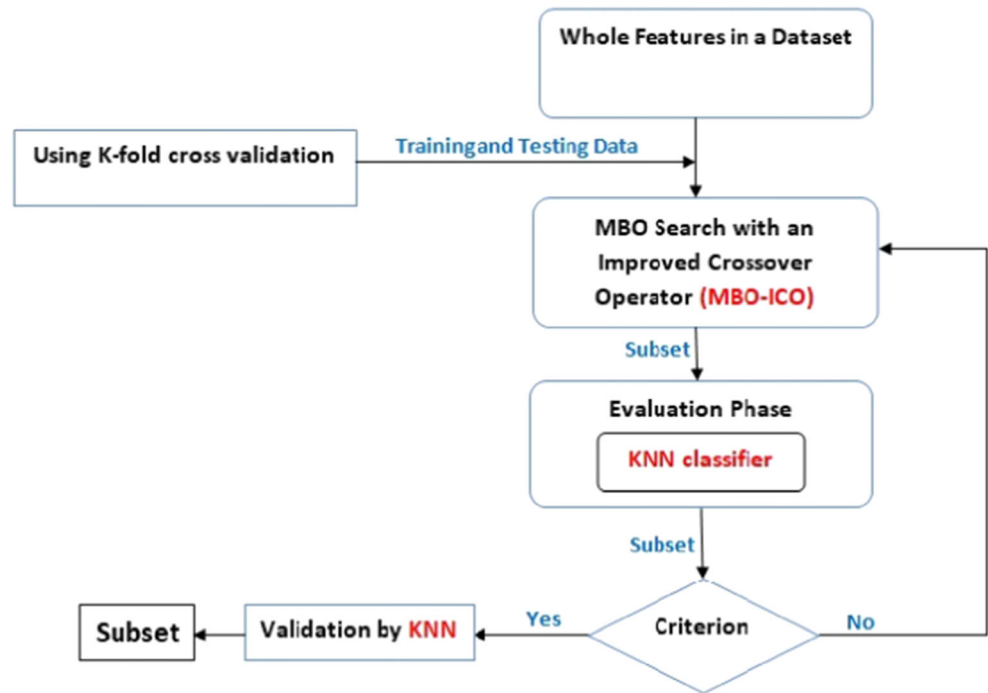
Animals and insects usually hunt for food randomly or semi-randomly, and have many delights in achieving their food target [83]. The next pathway in the hunt depends on the animal's current position and so on, and the path these

species may select depends on a mathematically designed model [84].

Recent research by Reynolds and Frye [85] have shown that many individuals explore nature on a sequence of normal pathways punctuated by a sudden 90o turn, leading to a discontinuous scale-free search pattern in the Lévy flight model. Only human conduct and light can be associated with flights to Lévy [86, 87].

Lévy flight is one of the strategies employed to increase the algorithm's convergence process and keep away from collapsing into local optima. This approach is first introduced by Lévy and Borel in 1954 [88]. In this process, the random walk 's phase length is extracted from a power law distribution with a long tail, so it is called Lévy distribution, and the new population is created near the current best solution, so it can speed up local search and avoid falling into local optimum [89].

Fig. 1 MBOICO with wrapper FS technique



Lévy flight reflects a random walk technique and its phase duration is calculated by the distribution of Lévy, which it considers to be a fast traileed distribution with a long path. It makes it easy to use in complicated exploration search cases and allows to escape from ideal local regions [90].

The random number generation in Lévy flight consists of two simple stages, the random choice of direction and the generation of stages that follow the Lévy chosen [91].

For similar random variables, the Lévy distribution can be described as the sum of variable numbers, and the Fourier transformation is defined as the following equation:

$$F(k) = \exp \left[-\alpha |k|^\beta \right], 0 < \beta \leq 2 \quad (14)$$

Where α is a scale parameter within $[-1, 1]$ interval and β is a Lévy index for random walk, and the step length S can be calculated by Mantegna's algorithm as:

$$S = \frac{\mu}{|v|^{1/\beta}} \quad (15)$$

Where μ and v are drawn from normal distributions. Which is $\mu \sim N(0, \sigma_\mu^2)$ and $v \sim N(0, \sigma_v^2)$. And σ_μ can be determined by the following equation:

$$\sigma_\mu = \left\{ \frac{r(1+b)\sin(\frac{\pi b}{2})^{1/b}}{r[(1+b)/2]b^{(b-1)/2}} \right\} \quad (16)$$

After that, the step size is calculated by:

$$\text{Stepsize} = 0.01 \times S \quad (17)$$

Where the 0.01 reflects the element that comes from the fact that $L/100$ would be the standard stage size of walks, and L is the traditional duration scale; otherwise, the hunt for Lévy flights may be too intense to allow new approaches outside the quest domain.

The velocity is modified using Lévy flight method to improve the efficiency of MBO quest. Initially initialize the full generation, the NP monarch butterfly populations randomly, peri , p , BAR , and S_{max} , close to simple MBO. Through using Lévy flight method to upgrade the MBO rpm, each butterfly takes a long leap toward improving butterfly diversity to improve MBO's global quest space exploration. In Lévy flight method parameter takes major role in distribution. By applying different values for β , the random distribution is changed differently. In this research, A constant value for β is chosen; $\beta = 1.5$.

Through integrating the benefits of random walk into the MBO, the butterfly positions in each iteration are strengthened. Algorithm 6 below shows the MBOLF

algorithm pseudocode while Fig. 2 shows the MBOLF algorithm flowchart.

Algorithm 6: The Pseudocode of MBOLF

- 1- **Begin**
 - 2- **Step 1: Initialization.**
 - 3- MaxGen the maximum generation, the population of
 - 4- monarch butterfly NP individuals randomly, peri, p, BAR, and Smax.
 - 5- **Step 2: While (G<MaxGen)**
 - 6- Sort the population according to their fitness.
 - 7- **If** rand() <0.5 **then**

 - 8- Change the velocity and position of practical by Eq.12 and 13
 - 9- **Else**
 - 10- Change the velocity and position of practical by Eq.1 and 2

 - 11- Produce sub-population1 with NP1 better individuals and hypothesis
 - 12- Sub-population2 on the remainders.
 - 13- Generate the offspring individual of sub-population1.
 - 14- Generate the offspring individual of sub-population2
 - 15- Recombine newly generated sub-populations into one population.
 - 16- Find current best solution.
 - 17- **Step 3:Find the global best solution .**

 - 18- **End.**
-

The suggested wrapper FS algorithm in this paper uses a KNN classifier as an evaluator and, at the same time, it uses the MBOLF as a search technique. The KNN classifier specifies the accuracy limit for the MBOLF chosen functions. Fig. 2 illustrates the procedures associated in using a wrapper FS strategy to implement the MBOLF.

Which is well known, the selection process is depends on two variables, namely the degree of precision and the numbers of selected features. To achieve a balance between these two variables, a fitness method for estimating the output was created. Eq. (9) reflects criteria for fitness function used in this paper:

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

where D is the rate of error classification obtained through KNN, R is the number of selected features, C is the number of features in the original data sets and β and α are the two parameters that denote the position of subset length and the classification quality, respectively, where $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ [67].

The level of complexity of the proposed method is dependent on the crossover operators, their application, the distribution of the individuals and the population, and the fitness function. In view of the normal choices (point mutation, point convergence, roulette wheel selection), the time complexity is $O(g(nm + nm + n))$, where mg is the number of generations, n is the population size and m is the individual size. So, the complexity is $O(gnm)$ [92, 93].

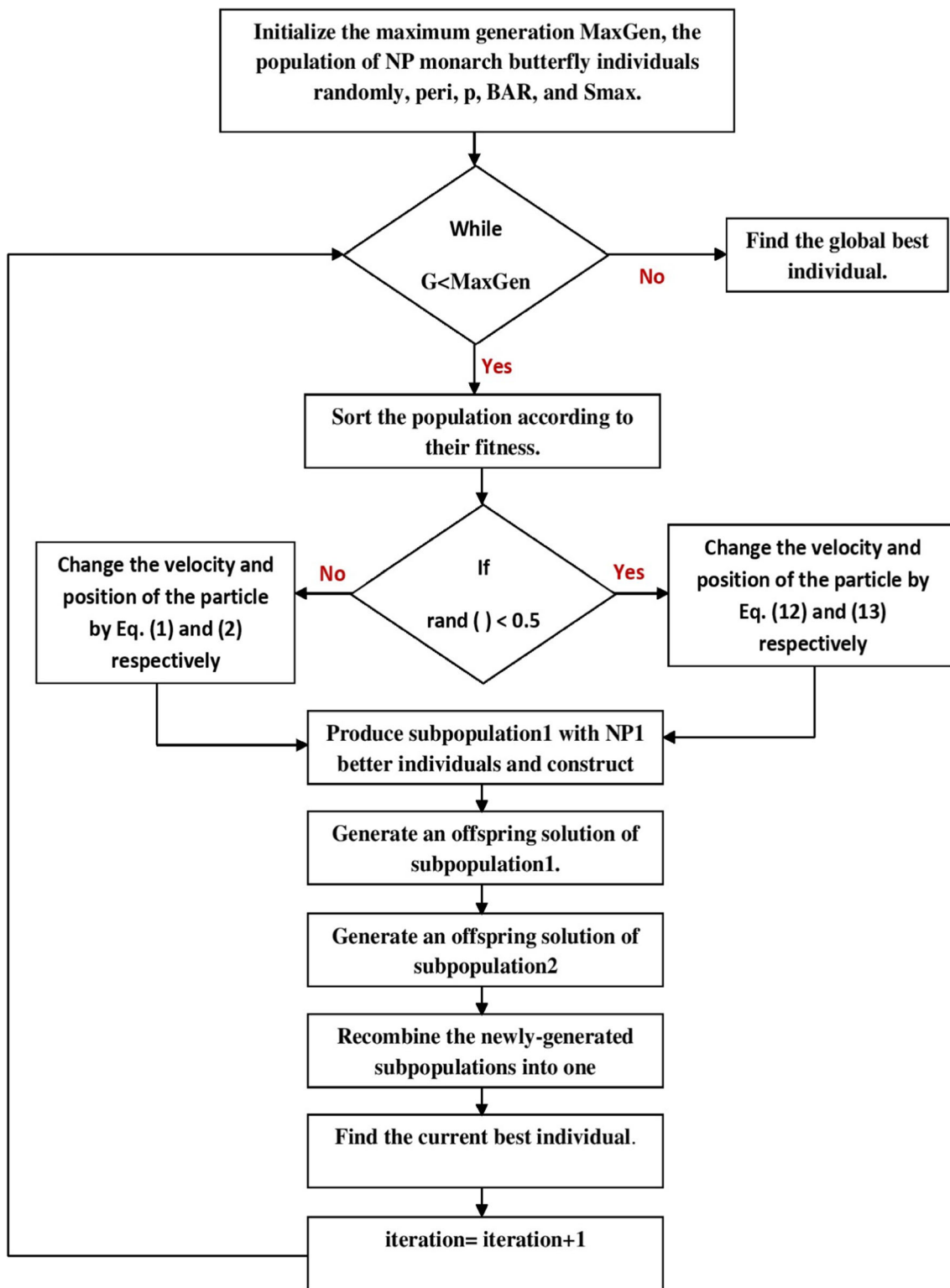


Fig. 2 The MBOLF Flowchart

The wrapper process, as stated earlier, embeds the classifier into the assessment phase to identify the proper features that need to be selected. This approach can produce results that are

more precise in terms of choosing the correct features to match the purpose of the analysis, but, at the same time it increases the amount of time taken to reach a solution. The wrapper

approach only tests $O((S + 1)N)$ candidate feature subsets to eventually choose S features and in the worst case $O(N^2)$ feature subsets [92].

4 Experiments

The wrapper FS model is integrated into the proposed methods using the assessment phase KNN classifier, where $K = 5$ is used to evaluate the accuracy of the selected features that MBO generates during the generation cycle. 25 UCI data warehouse comparison datasets were implemented to check the reliability of the proposed approach [46, 67, 68, 70]. Table 1 Shows the number of functions and objects within each of the data sets used.

Each dataset was split into a K -fold cross-validation training and testing data set, where $K-1$ folds were used for training and testing, and the remaining fold was used for testing.

Table 1 Dataset Details

	Dataset	No.of attributes	No.of objects
1.	Breastcancer	9	699
2.	BreastEW	30	569
3.	CongressEW	16	435
4.	Credit	20	1000
5.	Exactly	13	1000
6.	Exactly2	13	1000
7.	Derm	34	366
8.	Derm2	34	358
9.	HeartEW	13	270
10.	IonosphereEW	34	351
11.	KrvskpEW	36	3196
12.	LED	24	2000
13.	Lung	56	32
14.	Lymphography	18	148
15.	M-of-n	13	1000
16.	Mushroom	22	8124
17.	PenglungEW	325	73
18.	SonarEW	60	208
19.	SpectEW	22	267
20.	Tic-tac-toe	9	958
21.	Vote	16	300
22.	WaveformEW	40	5000
23.	WineEW	13	178
24.	WQ	12	4898
25.	Zoo	16	101

Table 2 Parameters Setting

Name of the Parameter	The Value
UB (upper bound).	1
LB (lower bound).	0
d-max (the max distance).	$Le-2$
Iterations number.	50
Size of the Population.	10
K (for KNN classifier).	5

The folds are chosen in such a way that each fold comprises approximately the same number of class marks. Considering the number of folds and repeats, KNN produces train/test splits such that the models of the same splits can be tested. The splits are provided as an ARFF file with the row ID, fold number, repeat number and class (TRAIN or TEST) as part of the job summary. The uploaded predictions are labelled with the test instance fold and repeat number, so that the effects can be correctly measured and aggregated. Instead KNN stores the results per fold/repeat as well as the aggregated points.

In the experiments the input parameters were set due to the results of some initial tests which allowed the proposed method to produce better performance. Table 2 displays the configurations for input parameters (Fig. 3).

4.1 The results and discussions

Three outcomes were taken into account in order to calculate the efficacy of the proposed methods: accuracy, number of selected features (selection size), and convergence speed. Table 3 displays the average accuracy and sample size performance of the two approaches suggested after 30 runs on each dataset.

MBOICO has obtained very good performance in terms of accuracy and number of selected features, as seen in Table 3. The average accuracy of this method in all datasets was 92%, which in the field of solving FS problems is known to be a very high rate, and the importance of such results is apparent as compared with the findings of other approaches in the literature discussed in the next section. The quality of the MBOICO's collected results is due to the crossover technique improved in MBO's quest approach, which allows the system to produce the best results possible while avoiding slipping into local minima.

MBOICO outperformed the MBOLF method in 14 datasets: CongressEW, Credit, Exactly2, Derm, IonosphereEW, KrvskpEW, Lung, M-of-n, SpectEW, Tic-tac-toe, Vote, WQ, WineEW and zoo. It produced the same results in terms of classification accuracy in six data sets: Breastcancer, Exactly, Derm2, Led, Mushroom and SonarEW.

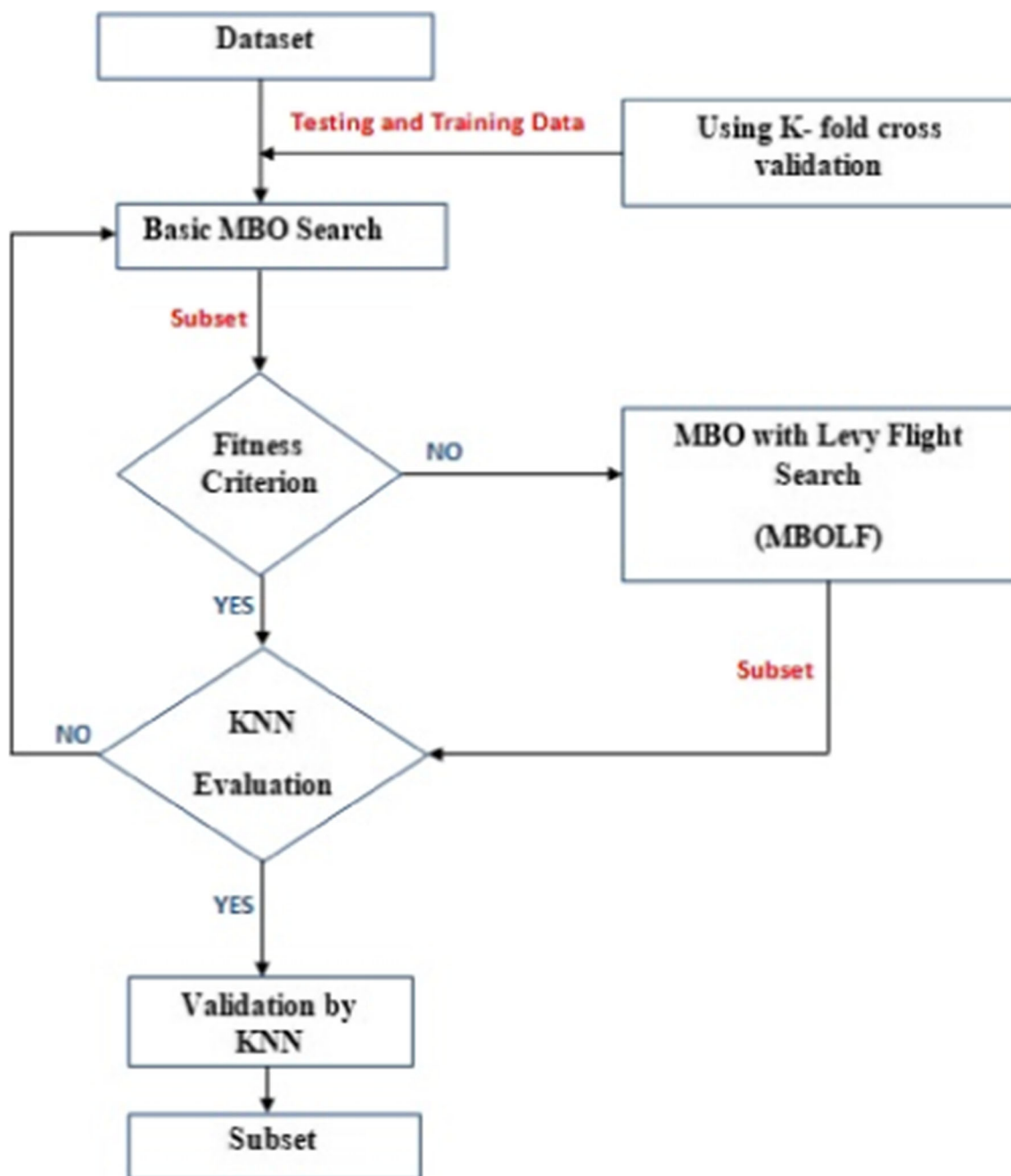


Fig. 3 MBOLF with wrapper FS technique

MBOLF also achieved high classification accuracy with an average of 91% in all data sets. It outperformed the MBOICO in five data sets: BreastEW, HeartEW, Lymphography, PenglungEW and WaveformEW. These high results of MBOLF obtained through using the mutation technique which made by Lévy flight.

As regards the number of features selected, the findings of the two approaches suggested have been very close;

MBOICO was the best in this term with an average of 12.04 selected features in all datasets, followed by MBOLF with an average of 12.40 in all datasets.

One of the goals of this work was to speed up optimization by reducing randomization and the exploration process by discovering the optimum solution in a shorter amount of time. From Fig. 4, we can note the MBOICO's success in identifying optimal solutions for

Table 3 MBOICO and MBOLF average accuracy and selection size

NO	Dataset	Accuracy		Features	
		MBOICO	MBOLF	MBOICO	MBOLF
1	Breastcancer	0.97	00.97	04.86	04.60
2	BreastEW	0.97	00.98	11.70	11.52
3	CongressEW	0.98	00.96	04.73	05.10
4	Credit	0.77	0.76	7.58	9.52
5	Exactly	1.00	01.00	06.00	07.00
6	Exactly2	0.77	00.75	02.53	03.12
7	Derm	0.79	0.78	10	10.89
8	Derm2	0.90	0.90	11.4	11.17
9	HeartEW	0.86	00.87	06.56	05.84
10	IonosphereEW	0.95	00.94	10.76	10.30
11	KrvskpEW	0.98	00.96	15.00	15.70
12	LED	0.99	0.99	5.14	5.07
13	Lung	0.99	0.97	10.06	11.40
14	Lymphography	0.90	00.91	07.56	07.14
15	M-of-n	1.00	00.99	06.00	08.20
16	Mushroom	1.00	1.00	5.23	5.86
17	PenglungEW	00.95	00.96	84.13	83.42
18	SonarEW	00.97	00.97	24.80	23.80
19	SpectEW	00.87	00.86	07.93	08.45
20	Tic-tac-toe	00.80	00.78	06.00	06.24
21	Vote	00.97	00.96	04.30	06.87
22	WaveformEW	00.78	00.79	19.00	20.15
23	WQ	0.80	0.78	9.81	10.16
24	WineEW	00.99	0.98	06.53	06.72
25	Zoo	00.99	00.97	05.63	05.86
	Average	0.92	0.91	12.01	12.40

25 data sets within the context of accelerated convergence.

Based on the results depicted in Fig. 4, the MBOLF method was the fastest algorithm in terms of finding better results within the search space. In all the datasets, it needed fewer than six iterations to improve the solution, except for the zoo dataset for which it needed 15 iterations to find the best solution. This confirms that the use of Lévy flight accelerates the convergence speed.

As for MBOICO method, although it achieved the best classification accuracy in most datasets, it could not improve the convergence speed to the same extent as MBOLF.

The basic MBO produced the worst results in terms of convergence speed, but this was expected because it was not modified in order to address the issue of speed, unlike MBOICO and MBOLF.

The above outcome is due to MBOICO and MBOLF achieving a good balance between local and global search in finding solutions, which prevents the search process from falling into local optima. The results also indicate that MBOICO and MBOLF start from a good initial state. Table 4 shows the number of iterations needed by each of the modified methods to get a better solution than the other method in the run time in all datasets.

Furthermore, the two methods, MBOICO and MBOLF, were both subjected to a T-test to assess the accuracy of all obtained data sets from each approach. The interval of importance was set at 95% ($\alpha = 0.05$). This research was undertaken to determine whether MBOICO's result was substantially different from the MBOLF's. Table 5 shows the findings for the MBOICO and the MBOLF, as well as the consistency *P*-values of the T-test. From the table it can be seen that the reliability of the MBOICO is very significant to MBA, as all *P*-values are less than 0.03.

4.2 Comparison of MBOICO with methods in the literature

MBOICO was used in this section for compatibility with approaches in the literature. This is preferred because it has the high accuracy of classification compared with the MBOLF system. The findings of the proposed MBOICO system were compared to the results of the FS methods, namely MBO, MBA-SA, WOASAT, BGSA and HGSA [46, 67, 68, 70]. All comparisons with the approaches in the literature were made using the same datasets and parameters as in those strategies. The distinction between the suggested MBOICO approach and the other methods was based on the following three criteria: accuracy of classification, FS size and fitness value. The MBOICO's classification accuracy and FS size were determined by taking the average accuracy and the average number of selected features, and measuring the mean, max and min fitness values. The fitness function and the accuracy were determined using Eq. (9).

The MBOICO method was initially compared with the MBA-SA method as both methods were applied to the same 25 datasets during the experimental phase to measure their efficiency. According to Table 6, MBOICO surpassed MBA-SA in 10 datasets: BreastEW, CongressEW, Credit, Exactly, Derm2, KrvskpEW, SonarEW, Tic-tac-toe, WaveformEW and WQ. Also, it produced the same results in nine datasets: Breastcancer, Derm, HeartEW, Lymphography, M-of-n, Mushroom, PenglungEW, WineEW and zoo. On the other hand, MBA-SA outperformed MBOICO in six datasets: Exactly2, IonosphereEW, Led, Lung, SpectEW and Vote. Overall, the average accuracy of

MBOICO was better than MBA-SA across all 25 datasets, at 92% versus 91%.

As for the size of the selected features, the superiority of the MBOICO approach is evident. It was able to surpass the

MBA-SA in 18 datasets. It also had a better average size of 12.02 features across all 25 datasets as opposed to an average of 15.97 for MBA-SA. The better overall performance of MBOICO is related to the modification which was applied

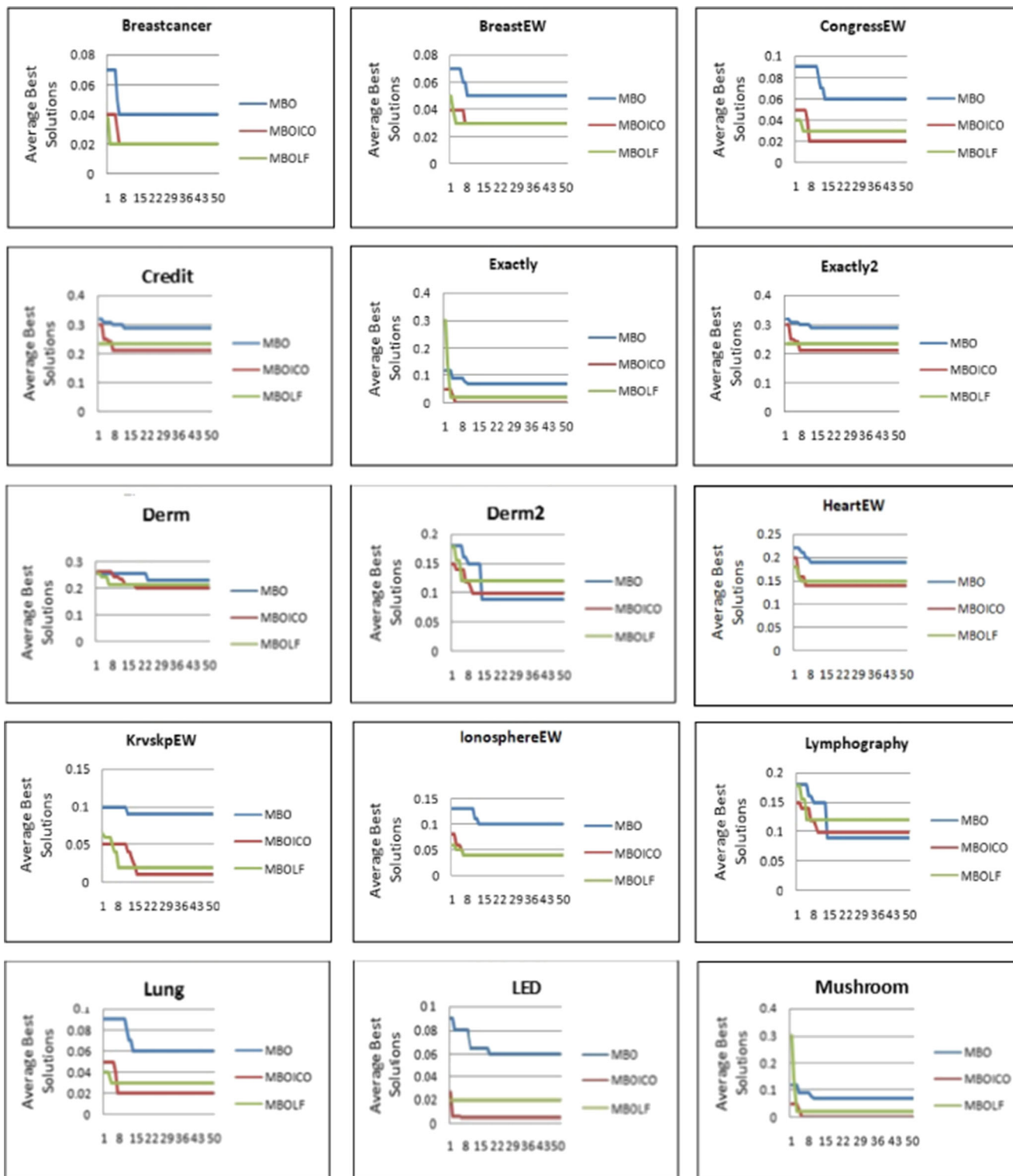


Fig. 4 The convergence speed of all methods

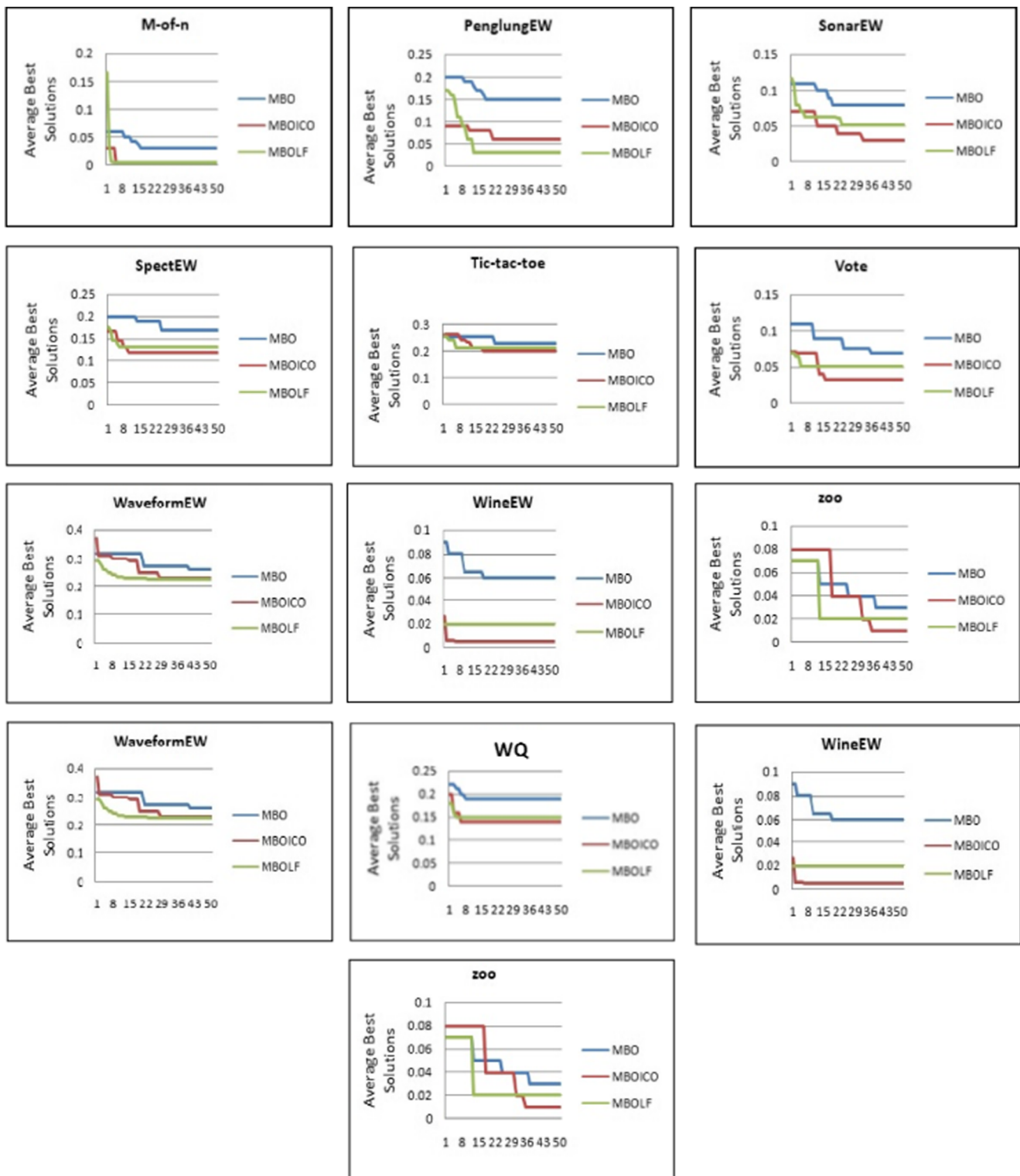


Fig. 4 (continued)

to the global search step to improve its ability to accurately select the most relevant features before progressing to local search.

Figure 5 shows the average of accuracy rate and the selection size of MBOICO and MBA-SA in all datasets used.

Table 4 The iteration number needed to achieve better solutions

NO	DATASETS	MBOICO	MBOLF
1	Breastcancer	5	2
2	BreastEW	7	2
3	CongressEW	6	3
4	Credit	7	1
5	Exactly	3	2
6	Exactly2	3	1
7	Derm	20	5
8	Derm2	8	4
9	HeartEW	3	3
10	IonosphereEW	3	3
11	KrvskpEW	12	2
12	Led	6	1
13	Lung	7	5
14	Lymphography	3	3
15	M-of-n	4	2
16	Mushroom	7	5
17	PenglungEW	12	3
18	SonarEW	12	2
19	SpectEW	4	2
20	Tic-tac-toe	8	3
21	Vote	14	3
22	WaveformEW	2	3
23	WQ	6	5
24	WineEW	2	1
25	Zoo	22	15

To confirm the reliability of the proposed method MBOICO and its capacity to provide such a high level of classification accuracy whilst at the same time effectively reducing the number features, its performance was also compared with that of four approaches in the literature, namely, MBO, WOASAT, BGSA and HGSA across 18 datasets that were used to test these methods in the original works. As seen in Table 7, and in contrast to WOASAT's test, MBOICO has demonstrated good success since it has received stronger outcomes. Moreover, in eight datasets it had the same outcomes, including Breastcancer, CongressEW, Exactly, KrvskpEW, M-of-n, SonarEW, Ballot, and WineEW. There were only three datasets where WOASAT reached MBOICO with an approximate very slight gap of 1%. These datasets were BreastEW, IonosphereEW and SpectEW. As regards the comparison with the results of BGSA, MBOICO obtained better results in nine datasets, namely, CongressEW, Exactly2, IonosphereEW, KrvskpEW, Lymphography, M-of-n, PenglungEW, SonarEW and Tic-tac-toe. However, in seven datasets, including Breastcancer, BreastEW, Exactly,

Table 5 P-values of the T-test

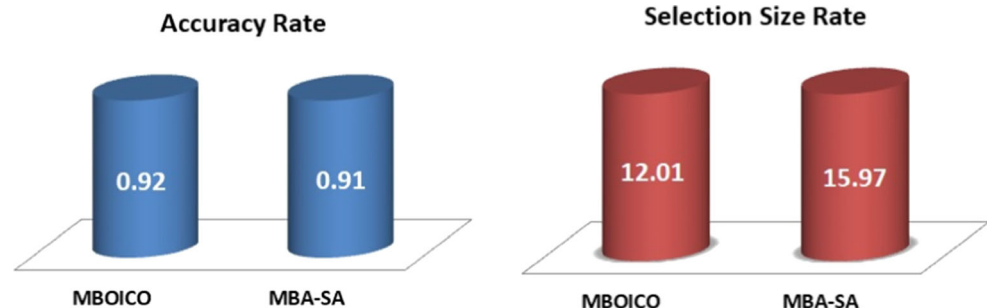
Dataset	Method	Mean	Std. deviation	Mean std. error	P value
Breastcancer	MBOICO	0.9756	0.01699	0.00421	0.00
	MBOLF	0.9701	0.02527	0.00129	
BreastEW	MBOICO	0.9798	0.02487	0.00394	0.01
	MBOLF	0.9802	0.01890	0.00301	
CongressEW	MBOICO	0.9860	0.04500	0.00788	0.00
	MBOLF	0.9644	0.01462	0.00205	
Credit	MBOICO	0.77418	0.04002	0.00614	0.00
	MBOLF	0.76037	0.01997	0.00351	
Exactly	MBOICO	1.0000	0.03514	0.00671	0.00
	MBOLF	1.0000	0.03789	0.00688	
Exactly2	MBOICO	0.7730	0.04442	0.00799	0.00
	MBOLF	0.7565	0.03039	0.00495	
Derm	MBOICO	0.79539	0.03510	0.00724	0.00
	MBOLF	0.78113	0.02104	0.00417	
Derm2	MBOICO	0.90814	0.05411	0.00851	0.00
	MBOLF	0.90007	0.02900	0.00600	
HeartEW	MBOICO	0.8687	0.05518	0.01197	0.01
	MBOLF	0.8701	0.02211	0.02478	
IonosphereEW	MBOICO	0.9574	0.02618	0.00490	0.00
	MBOLF	0.9404	0.02301	0.00512	
KrvskpEW	MBOICO	0.9897	0.06276	0.01109	0.00
	MBOLF	0.9616	0.02101	0.00310	
LED	MBOICO	0.99441	0.01721	0.00229	0.00
	MBOLF	0.99072	0.01400	0.00275	
Lung	MBOICO	0.99162	0.01052	0.00241	0.00
	MBOLF	0.97490	0.01173	0.00270	
Lymphography	MBOICO	0.9089	0.05248	0.00799	0.01
	MBOLF	0.9101	0.02100	0.00468	
M-of-n	MBOICO	1.0000	0.05358	0.00957	0.00
	MBOLF	0.9924	0.01650	0.00199	
Mushroom	MBOICO	1.0000	0.03227	0.00552	0.00
	MBOLF	1.0000	0.03431	0.00603	
PenglungEW	MBOICO	0.9589	0.04092	0.00940	0.02
	MBOLF	0.9607	0.00899	0.00299	
SonarEW	MBOICO	0.9779	0.03719	0.00669	0.00
	MBOLF	0.9727	0.01794	0.00288	
SpectEW	MBOICO	0.8788	0.03509	0.00617	0.00
	MBOLF	0.8611	0.02748	0.00497	
Tic-tac-toe	MBOICO	0.8068	0.04074	0.00690	0.00
	MBOLF	0.7809	0.01923	0.00297	
Vote	MBOICO	0.9745	0.03120	0.00539	0.00
	MBOLF	0.9618	0.02731	0.00524	
WaveformEW	MBOICO	0.7892	0.01812	0.6170	0.01
	MBOLF	0.7918	0.04800	0.7207	
WQ	MBOICO	0.80644	0.04171	0.00626	0.00
	MBOLF	0.78270	0.02006	0.00294	
WineEW	MBOICO	0.9974	0.03878	0.00622	0.00
	MBOLF	0.9827	0.01217	0.00174	
Zoo	MBOICO	0.9945	0.03336	0.00637	0.00
	MBOLF	0.9938	0.01536	0.00306	

HeartEW, Vote, WineEW and Zoo, MBOICO recorded the same findings as BGSA. There were only two datasets where BGSA surpassed MBOICO by an average very slight gap of 2.5%; these were SpectEW and WaveformEW. As for the contrast with HGSA tests, MBOICO obtained stronger performance in eight datasets,

Table 6 Comparison of MBOICO and MBA-SA by accuracy and feature selection size

Dataset	Accuracy		Feature size	
	MBOICO	MBA-SA	MBOICO	MBA-SA
1 Breastcancer	0.97	0.97	04.86	4.20
2 BreastEW	0.97	0.95	11.70	12.8
3 CongressEW	0.98	0.96	04.73	8.46
4 Credit	0.77	0.75	7.58	8.00
5 Exactly	1.00	0.90	06.00	7.66
6 Exactly2	0.77	0.79	02.53	3.26
7 Derm	0.79	0.79	10	10
8 Derm2	0.90	0.88	11.4	12.2
9 HeartEW	0.86	0.86	06.56	6.20
10 IonosphereEW	0.95	0.96	10.76	14.46
11 KrvskpEW	0.98	0.96	15.00	17.20
12 LED	0.99	1.00	5.14	6.1
13 Lung	0.99	1.00	10.06	11.4
14 Lymphography	0.90	0.90	07.56	8.30
15 M-of-n	1.00	1.00	06.00	7.70
16 Mushroom	1.00	1.00	5.23	4.1
17 PenglungEW	00.95	0.95	84.13	147.06
18 SonarEW	00.97	0.89	24.80	26.60
19 SpectEW	00.87	0.90	07.93	8.70
20 Tic-tac-toe	00.80	0.79	06.00	5.3
21 Vote	00.97	0.98	04.30	5.8
22 WaveformEW	00.78	0.77	19.00	20.10
23 WQ	0.80	0.78	9.81	12
24 WineEW	00.99	0.99	06.53	3.06
25 Zoo	00.99	0.99	05.63	2.86
Average	0.92	0.91	12.01	15.97

including CongressEW, HeartEW, IonosphereEW, KrvskpEW, Lymphography, SonarEW, Tic-tac-toe and

Fig. 5 Average accuracy and average feature selection size of MBOICO and MBA-SA**Table 7** Comparison of Accuracy of MBOICO and Methods in the Literature

NO.	Dataset	MBOICO	MBO	WOASAT	BGSA	HGSA
1	Breastcancer	00.97	00.96	00.97	00.97	00.97
2	BreastEW	00.97	00.95	00.98	00.97	00.97
3	CongressEW	00.98	00.94	00.98	00.96	00.96
4	Exactly	01.00	00.94	01.00	1.00	1.00
5	Exactly2	00.77	00.71	00.75	00.76	00.77
6	HeartEW	00.86	00.82	00.85	00.86	00.85
7	IonosphereEW	00.95	00.89	00.96	00.90	00.93
8	KrvskpEW	00.98	00.92	00.98	00.97	00.97
9	Lymphography	00.90	00.90	00.89	00.88	00.89
10	M-of-n	01.00	00.97	01.00	00.98	1.00
11	PenglungEW	00.95	00.84	00.94	00.93	00.95
12	SonarEW	00.97	00.94	00.97	00.89	00.96
13	SpectEW	00.87	00.83	00.88	00.89	00.92
14	Tic-tac-toe	00.80	00.77	00.79	00.79	00.79
15	Vote	00.97	00.93	00.97	00.97	00.97
16	WaveformEW	00.78	00.75	00.76	00.81	00.81
17	WineEW	00.99	00.94	00.99	00.99	00.99
18	Zoo	00.99	00.97	00.97	00.99	00.93
	Average	00.93	00.88	00.92	00.91	00.92

Zoo. Moreover, MBOICO provided the same findings as HGSA in eight datasets; namely, Breast cancer, BreastEW, Accurately, Exactly2, M-of-n, PenglungEW, Test, and WineEW. There were only two datasets where HGSA exceeded MBOICO with an average discrepancy of 4%; these were SpectEW and WaveformEW datasets. Ultimately, MBOICO beat all the other approaches across the 18 datasets, reaching an overall classification accuracy of 93%.

Figure 6 below also illustrates the superiority of the proposed approach MBOICO compared to the four methods in the literature, based on the average classification accuracy across all 18 datasets.

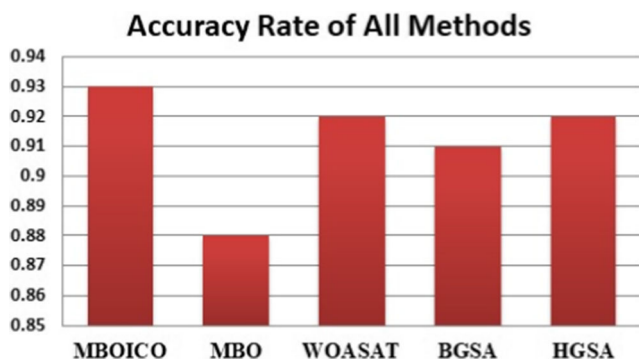


Fig. 6 Classification accuracy of all methods

Table 8 demonstrate the simple dominance of MBOICO in terms of reducing the number of features as comparing with the other algorithms. It surpassed the MBO in 11 datasets, and also exceeded WOASAT in 12 datasets. As well as MBOICO outperformed BGSA in 13 datasets and HGSA in eight datasets, and in a further eight datasets recorded the same findings as HGSA. It can be noted, the experiments on the 18 datasets showed that the MBOICO achieved a higher selection size rate than all the other methods except HGSA, and it was 13 features in average.

Figure 5 also demonstrates the superiority of the proposed approach over all previous methods except for HGSA.

This work also examined the results during the run cycle of the proposed algorithm to determine the best, bad and average fitness values for all the data sets evaluated. Table 9 displays the best fitness values collected for each sample during 30 MBOICO tests, and contrasts the results with those of the literature's four processes. MBOICO and WOASAT have the highest fitness with a mean value of 0.03, as can be seen from the table. The lowest fitness metrics for both strategies are displayed in Table 10. This can be shown that for all datasets MBA-SA obtained the worst overall performance, followed respectively by the MBO, MBOICO and WOASAT. As for the average fitness values, Table 11 reveals that of all the approaches, MBOICO and HGSA had the highest average fitness (Fig. 7).

From the above analyses, the MBOICO achieved a high level of performance in terms of accurately selecting the best features and in selecting a relatively small number of features (selection size) as compared to the other tested algorithms. The balance that it achieved between global and local search was clearly demonstrated by the fast convergence speed. This shows that the proposed method has great potential for use in finding integrated and ideal solutions for applications that depend on a fast search process. In this paper, the global search was updated to improve its productivity by finding the right points from the start and before going on to local search. Thus this methodological approach differs from other approaches in the literature because it

Table 8 Selection Size Rate by All Methods

NO.	Dataset	MBOICO	MBO	WOASAT	BGSA	HGSA
1	Breastcancer	04.86	05.00	04.20	4.1	4.0
2	BreastEW	11.70	11.20	11.60	16.2	13.8
3	CongressEW	04.73	04.00	06.40	5.4	3.9
4	Exactly	06.00	07.40	06.00	6.0	6.0
5	Exactly2	02.53	03.52	02.80	3.5	4.7
6	HeartEW	06.56	06.30	05.40	6.4	6.7
7	IonosphereEW	10.76	12.00	12.80	10.9	8.2
8	KrvskpEW	15.00	14.23	18.40	17.0	15.7
9	Lymphography	07.56	8.00	07.20	7.4	7.0
10	M-of-n	06.00	09.54	06.00	7.0	6.0
11	PenglungEW	84.13	81.78	127.40	150.1	43.0
12	SonarEW	24.80	26.20	26.40	26.0	25.4
13	SpectEW	07.93	08.13	09.40	11.6	8.0
14	Tic-tac-toe	06.00	05.64	06.00	8.9	8.7
15	Vote	04.30	07.00	05.20	9.0	3.1
16	WaveformEW	19.00	22.98	20.60	20.0	19.5
17	WineEW	06.53	07.48	06.40	6.2	5.0
18	Zoo	05.63	05.69	05.60	7.8	5.6
	Average	13.00	13.67	15.98	18.5	11.1

Table 9 Best Fitness Values Produced by All Methods

NO.	Dataset	MBOICO	MBO	MBA-SA	WOASAT	BGSA	HGSA
1	Breastcancer	00.02	00.02	0.01	00.03	–	–
2	BreastEW	00.01	00.02	0.02	00.02	–	–
3	CongressEW	00.00	00.01	0.02	00.02	–	–
4	Exactly	00.00	00.00	0.05	00.01	–	–
5	Exactly2	00.10	00.20	0.18	00.02	–	–
6	HeartEW	00.10	00.12	0.09	00.13	–	–
7	IonosphereEW	00.02	00.03	0.02	00.03	–	–
8	KrvskpEW	00.02	00.02	0.02	00.02	–	–
9	Lymphography	00.04	00.05	0.06	00.09	–	–
10	M-of-n	0.00	0.00	0.00	0.01	–	–
11	PenglungEW	00.02	00.02	0.01	00.03	–	–
12	SonarEW	00.00	00.01	0.08	00.01	–	–
13	SpectEW	00.10	00.10	0.06	00.11	–	–
14	Tic-tac-toe	00.10	00.17	0.15	00.20	–	–
15	Vote	00.00	00.00	0.01	00.02	–	–
16	WaveformEW	00.10	00.20	0.18	00.23	–	–
17	WineEW	00.00	00.00	0.00	00.00	–	–
18	Zoo	00.00	00.00	0.00	00.00	–	–
	Average	00.03	00.05	0.05	00.05	–	–

Table 10 Worst Fitness Values Produced by All Methods

NO.	Dataset	MBOICO	MBO	MBA-SA	WOASAT	BGSA	HGSA
1	Breastcancer	00.04	00.05	0.04	00.04	–	–
2	BreastEW	00.05	00.04	0.07	00.04	–	–
3	CongressEW	00.02	00.03	0.07	00.05	–	–
4	Exactly	00.00	00.00	0.16	00.01	–	–
5	Exactly2	00.21	00.27	0.23	00.27	–	–
6	HeartEW	00.12	00.16	0.19	00.18	–	–
7	IonosphereEW	00.08	00.08	0.09	00.05	–	–
8	KrvskpEW	00.01	00.02	0.08	00.02	–	–
9	Lymphography	00.10	00.15	0.17	00.14	–	–
10	M-of-n	00.02	00.00	0.09	00.01	–	–
11	PenglungEW	00.13	00.16	0.14	00.11	–	–
12	SonarEW	00.10	00.09	0.15	0.05	–	–
13	SpectEW	00.23	00.20	0.15	00.15	–	–
14	Tic-tac-toe	00.27	00.22	0.25	00.23	–	–
15	Vote	00.03	00.05	0.5	00.04	–	–
16	WaveformEW	00.18	00.25	0.25	00.26	–	–
17	WineEW	00.02	00.03	0.03	00.03	–	–
18	Zoo	00.09	00.09	0.05	00.10	–	–
	Average	00.09	00.10	0.15	00.09	–	–

Table 11 Average Fitness Values Produced by All Methods

NO.	Dataset	MBOICO	MBO	MBA-SA	WOASAT	BGSA	HGSA
1	Breastcancer	00.03	–	0.03	00.04	0.03	0.03
2	BreastEW	00.03	–	0.05	00.03	0.03	0.03
3	CongressEW	00.02	–	0.04	00.03	0.03	0.03
4	Exactly	00.00	–	0.10	00.01	0.005	0.00
5	Exactly2	00.23	–	0.21	00.25	0.23	0.23
6	HeartEW	00.14	–	0.14	00.16	0.13	0.014
7	IonosphereEW	00.05	–	0.04	00.04	0.09	0.06
8	KrvskpEW	00.02	–	0.04	00.02	0.03	0.02
9	Lymphography	00.10	–	0.10	00.11	0.11	0.11
10	M-of-n	00.00	–	0.00	00.01	0.02	0.00
11	PenglungEW	00.05	–	0.05	00.06	0.06	0.04
12	SonarEW	00.03	–	0.11	00.03	0.11	0.04
13	SpectEW	00.13	–	0.10	00.13	0.11	0.08
14	Tic-tac-toe	00.20	–	0.21	00.21	0.21	0.22
15	Vote	00.03	–	0.02	00.04	0.03	0.02
16	WaveformEW	00.22	–	0.22	00.25	0.18	0.18
17	WineEW	00.01	–	0.01	00.01	0.01	0.01
18	Zoo	00.01	–	0.01	00.04	0.00	0.07
	Average	00.07	–	0.08	00.08	0.08	0.07

reveals the significance of global search in solving FS problems.

5 Conclusion

Feature selection is a complicated task in the fields of data mining and machine learning. Researchers have been struggling for some years to find the best way to identify the lowest total number of features so that the initial dataset can be represented as precisely and as easily as

possible. Recent studies have focused on using metaheuristic algorithms to select the required features. One of these algorithms is the MBO algorithm, which was subjected to two modifications in this study: 1) the addition of an improved crossover operator (MBOICO) and 2) the incorporation of Lévy flight (MBOLF). These two modified methods were then applied with a wrapper FS system and KNN classifier to 25 benchmark data sets. In the development and evaluation phases, the two approaches used the KNN classifier. The two modified methods suggested were also tested on 18 other data sets in order to compare their results with those of four models in the literature (BGSA, HGSA, WOASAT, MBO) and MBA-SA. The findings demonstrated the MBOICO's dominance in terms of accuracy and number of selected features, where it achieved an overall accuracy rate of 93% across 25 datasets, which can be regarded as a quite clear victory in this field of study. However, there is still room for further improvement. Therefore, in future works, researchers may wish to employ an SVM instead of KNN, and they may also wish to hybridize the MBO algorithm with other metaheuristic algorithms as well as another classifier. Furthermore, the modified MBO has the potential to be applied as an FS search technique in other fields such as the internet of things, image segmentation and sentiment analysis among others.

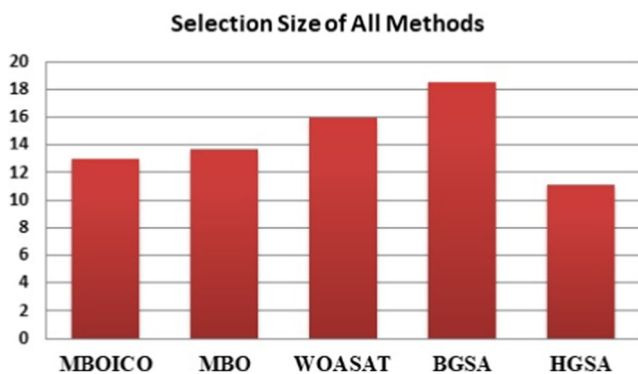


Fig. 7 Number of selected features by all methods

Acknowledgements The work published in this paper was funded by Al-Balqa Applied University, The Deanship of Scientific Research, Al-Salt, Jordan.

References

- Niu X, Wang J (2019) A combined model based on data preprocessing strategy and multi-objective optimization algorithm for short-term wind speed forecasting. *Appl Energy* 241:519–539
- Masoudi-Sobhanzadeh Y, Motieghader H, Masoudi-Nejad A (2019) FeatureSelect: a software for feature selection based on machine learning approaches. *BMC Bioinf* 20(1):170
- Sayed S, Nassef M, Badr A, Farag I (2019) A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. *Expert Syst Appl* 121:233–243
- Alweshah M, Qadoura MA, Hammouri AI, Azmi MS, AlKhalailah S (2020) Flower Pollination Algorithm for Solving Classification Problems. *Int. J. Advance Soft Compu. Appl* 12(1)
- Al Nsour H, Alweshah M, Hammouri AI, Al Ofeishat H, Mirjalili S (2018) A hybrid grey wolf optimiser algorithm for solving time series classification problems. *J Intell Syst* 29(1):846–857
- Mohamed AW, Hadi AA, Jambi KM (2019) Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. *Swarm and Evolutionary Computation* 50:100455
- Manoj RJ, Praveena MA, Vijayakumar K (2019) An ACO–ANN based feature selection algorithm for big data. *Clust Comput* 22(2):3953–3960
- Alweshah M, Al-Sendah M, Dorgham OM, Al-Momani A, Tedmori S (2020) Improved water cycle algorithm with probabilistic neural network to solve classification problems. *Clust Comput*: 1–16
- M Alweshah, L Rababa, MH Ryalat, A Al Momani, and MF Ababneh (2020). “African Buffalo algorithm: training the probabilistic neural network to solve classification problems,” *Journal of King Saud University-Computer and Information Sciences*
- M Alweshah, A Al-Daradkeh, MA Al-Betar, A Almomani, and S Oqeili (2019). “ β -hill climbing algorithm with probabilistic neural network for classification problems,” *J Ambient Intell Humaniz Comput*, pp. 1–12
- Hashemi A, Dowlatshahi MB, Nezamabadi-pour H (2020) MGFS: a multi-label graph-based feature selection algorithm via PageRank centrality. *Expert Syst Appl* 142:113024
- H Djellali, NG Zine, and N Azizi (2016). “Two stages feature selection based on filter ranking methods and SVMRF on medical applications,” In *Modelling and Implementation of Complex Systems*: Springer, pp. 281–293
- Hancer E, Xue B, Zhang M (2018) Differential evolution for filter feature selection based on information theory and feature ranking. *Knowl-Based Syst* 140:103–119
- Apolloni J, Leguizamón G, Alba E (2016) Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Appl Soft Comput* 38:922–932
- Liu H, Sun J, Liu L, Zhang H (2009) Feature selection with dynamic mutual information. *Pattern Recogn* 42(7):1330–1339
- Karegowda AG, Manjunath A, Jayaram M (2010) Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management* 2(2):271–277
- Uğuz H (2011) A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowl-Based Syst* 24(7):1024–1032
- J Tang, S Alelyani, and H Liu (2014). “Feature selection for classification: A review,” *Data classification: Algorithms and applications*, p. 37
- Rodriguez-Galiano V, Luque-Espinar J, Chica-Olmo M, Mendes M (2018) Feature selection approaches for predictive modelling of groundwater nitrate pollution: an evaluation of filters, embedded and wrapper methods. *Sci Total Environ* 624:661–672
- Yusta SC (2009) Different metaheuristic strategies to solve the feature selection problem. *Pattern Recogn Lett* 30(5):525–534
- Wang J, Hedar A-R, Wang S, Ma J (2012) Rough set and scatter search metaheuristic based feature selection for credit scoring. *Expert Syst Appl* 39(6):6123–6128
- Wahono RS, Suryana N, Ahmad S (2014) Metaheuristic optimization based feature selection for software defect prediction. *Journal of Software* 9(5):1324–1333
- Alweshah M, Rashaideh H, Hammouri AI, Tayyeb H, Ababneh M (2017) Solving time series classification problems using support vector machine and neural network. *International journal of data analysis techniques and strategies* 9(3):237–247
- SR Ahmad, AA Bakar, and MR Yaakub (2015). “Metaheuristic algorithms for feature selection in sentiment analysis,” In *2015 Science and Information Conference (SAI)*, pp. 222–226: IEEE
- M Alweshah, E Ramadan, MH Ryalat, M Almi’ani, and AI Hammouri (2020). “Water Evaporation Algorithm With Probabilistic Neural Network For Solving Classification Problems,” *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 6, no. 01
- Gokalp O, Tasci E, Ugur A (2020) A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification. *Expert Syst Appl* 146:113176
- Alweshah M, Abdullah S (2015) Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems. *Appl Soft Comput* 35:513–524
- Alweshah M, Alzubi OA, Alzubi JA, Alaqeel S (2016) Solving attribute reduction problem using wrapper genetic programming. *International Journal of Computer Science and Network Security (IJCSNS)* 16(5):77
- Alweshah M (2019) Construction biogeography-based optimization algorithm for solving classification problems. *Neural Comput & Applic* 31(10):5679–5688
- X-S Yang (2010). *Engineering optimization: an introduction with metaheuristic applications*. Wiley
- Alweshah M, Hammouri AI, Tedmori S (2017) Biogeography-based optimisation for data classification problems. *International Journal of Data Mining, Modelling and Management* 9(2):142–162
- Sayed GI, Tharwat A, Hassanien AE (2019) Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. *Appl Intell* 49(1):188–205
- Aghdam MH, Ghasem-Aghaee N, Basiri ME (2009) Text feature selection using ant colony optimization. *Expert Syst Appl* 36(3):6843–6853
- Sakri SB, Rashid NBA, Zain ZM (2018) Particle swarm optimization feature selection for breast cancer recurrence prediction. *IEEE Access* 6:29637–29647
- Shunmugapriya P, Kanmani S (2017) A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC hybrid). *Swarm and Evolutionary Computation* 36:27–36
- Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, al-Zoubi A’M, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl-Based Syst* 145:25–45
- Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. *Neural Comput & Applic* 31(1):171–188

38. Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. *Inf Sci* 497:219–239
39. Lin K-C, Zhang K-Y, Huang Y-H, Hung JC, Yen N (2016) Feature selection based on an improved cat swarm optimization algorithm for big data classification. *J Supercomput* 72(8):3210–3221
40. Diao R, Shen Q (2012) Feature selection with harmony search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42(6):1509–1523
41. D Rodrigues et al. (2013). “BCS: A binary cuckoo search algorithm for feature selection.” In 2013 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 465–468: IEEE
42. Alweshah M, Al-Daradkeh A, Al-Betar MA, Almomani A, Oqeili S (2019) β -hill climbing algorithm with probabilistic neural network for classification problems. *J Ambient Intell Humaniz Comput*:1–12
43. Al Nsour H, Alweshah M, Hammouri AI, Al Ofeishat H, Mirjalili S (2019) A hybrid Grey wolf Optimiser algorithm for solving time series classification problems. *J Intell Syst*
44. Mohammed Alweshah, Moad Abu Qadoura, Abdelaziz I. Hammouri, Mohd Sanusi Azmi, and S Alkhalaileh (2019). “Flower Pollination Algorithm for solving classification problems,” *International Journal of Advances in Soft Computing and Its Applications*, vol. In Press, pp. 1–13
45. Wang G-G, Deb S, Cui Z (2019) Monarch butterfly optimization. *Neural Comput & Applic* 31(7):1995–2014
46. Alweshah M, Al Khalaileh S, Gupta BB, Almomani A, Hammouri AI, Al-Betar MA (2020) The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Comput & Applic*:1–15
47. Wang G-G, Deb S, Coelho LDS (2018) Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. *International Journal of Bio-Inspired Computation* 12(1):1–22
48. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl Intell* 48(10):3462–3481
49. Chatterjee B, Bhattacharyya T, Ghosh KK, Singh PK, Geem ZW, Sarkar R (2020) Late acceptance hill climbing based social ski driver algorithm for feature selection. *IEEE Access* 8:75393–75408
50. M Ghosh, S Malakar, S Bhowmik, R Sarkar, and M Nasipuri (2019). “Feature Selection for Handwritten Word Recognition Using Memetic Algorithm,” In *Advances in Intelligent Computing*: Springer, pp. 103–124
51. HM Zawbaa, E Emary, and B Parv (2015). “Feature selection based on antlion optimization algorithm,” in *Complex Systems (WCCS), 2015 Third World Conference on*, pp. 1–7: IEEE
52. C Shivalingegowda and P Jayasree (2020). “Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks,” *J Ambient Intell Humaniz Comput*, pp. 1–14
53. Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VHC, Mirjalili S (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst Appl* 139:112824
54. S Wu (2015). “Comparative analysis of particle swarm optimization algorithms for text feature selection.”
55. Zawbaa HM, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. *PLoS one* 11(3):e0150652
56. J Wang, B Xue, X Gao, and M Zhang (2016). “A differential evolution approach to feature selection and instance selection,” In *Pacific Rim International Conference on Artificial Intelligence*, pp. 588–602: Springer
57. Shahbeig S, SADJAD K, SADEGHI M (2016) Feature selection from Iron direct reduction data based on binary differential evolution optimization. *Bulletin de la Société Royale des Sciences de Liège* 85:114–122
58. Malakar S, Ghosh M, Bhowmik S, Sarkar R, Nasipuri M (2019) A GA based hierarchical feature selection approach for handwritten word recognition. *Neural Comput & Applic*:1–20
59. R Saidi, W Bouaguel, and N Essoussi (2019). “Hybrid Feature Selection Method Based on the Genetic Algorithm and Pearson Correlation Coefficient,” In *Machine Learning Paradigms: Theory and Application*: Springer, pp. 3–24
60. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. *Neurocomputing* 213:54–65
61. ME Basiri and S Nemati (2009). “A novel hybrid ACO-GA algorithm for text feature selection,” In *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 2561–2568: IEEE
62. Jona J, Nagaveni N (2014) Ant-cuckoo colony optimization for feature selection in digital mammogram. *Pakistan journal of biological sciences: PJBS* 17(2):266–271
63. Mafarja M, Abdullah S (2013) Investigating memetic algorithm in solving rough set attribute reduction. *Int J Comput Appl Technol* 48(3):195–202
64. Zorarpaci E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Syst Appl* 62:91–103
65. Zhang Y, Gong D-w, Cheng J (2015) Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM transactions on computational biology and bioinformatics* 14(1):64–75
66. Zhang Y, Gong D, Hu Y, Zhang W (2015) Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing* 148:150–157
67. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312
68. Mafarja M, Aljarah I, Faris H, Hammouri AI, Ala’M A-Z, Mirjalili S (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst Appl* 117:267–286
69. Faris H, Mafarja MM, Heidari AA, Aljarah I, al-Zoubi A’M, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl-Based Syst* 154:43–67
70. Alweshah M, Alkhalaileh S, Albashish D, Mafarja M, Bsoul Q, Dorgham O (2020) A hybrid mine blast algorithm for feature selection problems. *Soft Comput*:1–18
71. Feng Y, Yang J, Wu C, Lu M, Zhao X-J (2018) Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian mutation. *Memetic Computing* 10(2):135–150
72. S Chakrabarty, AK Pal, N Dey, D Das, and S Acharjee (2014). “Foliage area computation using Monarch butterfly algorithm,” In *Non Conventional Energy (ICONCE), 2014 1st International Conference on*, pp. 249–253: IEEE
73. Ghanem WA, Jantan A (2018) Hybridizing artificial bee colony with monarch butterfly optimization for numerical optimization problems. *Neural Comput & Applic* 30(1):163–181
74. Devikanniga D, Raj RJS (2018) Classification of osteoporosis by artificial neural network based on monarch butterfly optimisation algorithm. *Healthcare technology letters* 5(2):70–75
75. I Strumberger, M Sarac, D Markovic, and N Bacanin (2018). “Hybridized Monarch Butterfly Algorithm for Global Optimization Problems,” *International Journal of Computers*, vol. 3
76. Faris H, Aljarah I, Mirjalili S (2018) Improved monarch butterfly optimization for unconstrained global search and neural network training. *Appl Intell* 48(2):445–464
77. Yazdani S, Hadavandi E (2018) LMBO-DE: a linearized monarch butterfly optimization algorithm improved with differential evolution. *Soft Comput*:1–15

78. I Stromberger, E Tuba, N Bacanin, M Beko, and M Tuba (2018). "Monarch butterfly optimization algorithm for localization in wireless sensor networks," In *Radioelektronika (RADIOELEKTRONIKA)*, 2018 28th International Conference, pp. 1–6: IEEE
79. G-G Wang, X Zhao, and S Deb (2015). "A novel monarch butterfly optimization with greedy strategy and self-adaptive," In *Soft Computing and Machine Intelligence (ISCFI)*, 2015 Second International Conference on, pp. 45–50: IEEE
80. Feng Y, Wang G-G, Li W, Li N (2018) Multi-strategy monarch butterfly optimization algorithm for discounted {0-1} knapsack problem. *Neural Comput & Applic* 30(10):3019–3036
81. G-G Wang, X Zhao, and S Deb (2015). "A novel monarch butterfly optimization with greedy strategy and self-adaptive," in 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCFI), pp. 45–50: IEEE
82. Wang G-G, Deb S, Zhao X, Cui Z (2018) A new monarch butterfly optimization with an improved crossover operator. *Oper Res* 18(3): 731–755
83. Brown CT, Liebovitch LS, Glendon R (2007) Lévy flights in Dobe Ju/'hoansi foraging patterns. *Hum Ecol* 35(1):129–138
84. Pavlyukevich I (2007) Lévy flights, non-local search and simulated annealing. *J Comput Phys* 226(2):1830–1844
85. Reynolds AM, Frye MA (2007) Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search. *PloS one* 2(4):e354
86. Barthelemy P, Bertolotti J, Wiersma DS (2008) A Lévy flight for light. *Nature* 453(7194):495
87. Shlesinger MF (2006) Mathematical physics: Search research. *Nature* 443(7109):281
88. P Lévy (1954). *Théorie de l'addition des variables aléatoires*. Gauthier-Villars
89. He X, Huang J, Rao Y, Gao L (2016) Chaotic teaching-learning-based optimization with Lévy flight for global numerical optimization. *Computational intelligence and neuroscience* 2016:43
90. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
91. Jensi R, Jiji GW (2016) An enhanced particle swarm optimization with levy flight for global optimization. *Appl Soft Comput* 43:248–261
92. H Liu and H Motoda (2007). *Computational methods of feature selection*. CRC Press
93. He J, Yao X (2001) Drift analysis and average time complexity of evolutionary algorithms. *Artif Intell* 127(1):57–85

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dr. Mohammed Alweshah received his BS in computer sciences in 1993, from Al-Mustansiriah University, Iraq and his master degree of computer sciences in 2005, from Al-Balqa Applied University, Jordan. Mohammed Alweshah received his Ph.D. degree from the Computer Science Department, School of Information Technology, National University of Malaysia (UKM), Malaysia in 2013. He was working under the supervision of Prof. Salwani

Abdullah. Currently, Alweshah is a faculty member with the Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University, AlSalt-Jordan since 2005. His research interests include Heuristics and Meta-heuristics, Evolutionary Algorithms, and data mining problems.