Applied Soft Computing 62 (2018) 441–453

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Full Length Article

Whale optimization approaches for wrapper feature selection

Majdi Mafarja [a,*], Seyedali Mirjalili [b]

[a] Department of Computer Science, Birzeit University, POBox 14, West Bank, Palestine
[b] Institute for Integrated and Intelligent Systems, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia

ARTICLE INFO

Article history:
Received 20 September 2016
Received in revised form 30 October 2017
Accepted 3 November 2017
Available online 8 November 2017

Keywords:
Feature selection
Optimization
Whale optimization algorithm
Classification
WOA
Crossover
Mutation
Selection
Evolutionary operators

ABSTRACT

Classification accuracy highly dependents on the nature of the features in a dataset which may contain irrelevant or redundant data. The main aim of feature selection is to eliminate these types of features to enhance the classification accuracy. The wrapper feature selection model works on the feature set to reduce the number of features and improve the classification accuracy simultaneously. In this work, a new wrapper feature selection approach is proposed based on Whale Optimization Algorithm (WOA). WOA is a newly proposed algorithm that has not been systematically applied to feature selection problems yet. Two binary variants of the WOA algorithm are proposed to search the optimal feature subsets for classification purposes. In the first one, we aim to study the influence of using the Tournament and Roulette Wheel selection mechanisms instead of using a random operator in the searching process. In the second approach, crossover and mutation operators are used to enhance the exploitation of the WOA algorithm. The proposed methods are tested on standard benchmark datasets and then compared to three algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), the Ant Lion Optimizer (ALO), and five standard filter feature selection methods. The paper also considers an extensive study of the parameter setting for the proposed technique. The results show the efficiency of the proposed approaches in searching for the optimal feature subsets.

1. Introduction

Due to the huge increase in the information amount in the world, a pre-processing technique like feature selection becomes a necessary and challenging step when using a data mining technique [1]. Feature selection can be defined as the process of eliminating the redundant and irrelevant features from a dataset to enhance the learning algorithm in the following steps [2]. Feature selection methods can be classified based on two criteria: the search strategy and the evaluation criterion [2]. The selected subset of features could be evaluated using two different techniques: filter and wrapper. A wrapper-based approach includes a learning algorithm (*e.g.* classification algorithm) in the selection process [3]. On the other hand, in the filter approach, the feature space will be searched depending on the data itself rather than on the learning method (e.g., classifier in a wrapper method [4–8]. In literature, there are many studies that conduct a comparison between many feature selection algorithms as in [9] where eight standard feature selection methods have been tested and evaluated using three different classifiers.

Searching an optimal subset is a crucial issue in feature selection. All possible subsets could be generated in an exhaustive manner to generate the best subset [10]. This approach is inapplicable for the huge datasets because if a dataset contains N features, then 2^N solutions should be generated and evaluated resulting in an extremely high computational cost [11]. Another approach is to search for a minimal reduct randomly [12]. The best case in that approach is when finding the minimal reduct in early stages while the worst case of this approach is that it might perform a complete search. Furthermore, the heuristic, as the name suggests, uses heuristic information to guide the search. Although a heuristic search strategy does not guarantee finding the best subset, it normally finds an acceptable solution in a reasonable time [13]. Heuristic methods may be classified into two families: specific heuristics developed to solve a certain problem and general-purposed metaheuristics to solve a wide range of problems [13]. In the past two decades, metaheuristics have showed their efficiency and effectiveness in solving challenging and large-scale problems in engineering design, machine learning, data mining scheduling, and production problems.

* Corresponding author.
E-mail addresses: mmafarja@birzeit.edu, mmafarjeh@gmail.com (M. Mafarja), seyedali.mirjalili@griffithuni.edu.au (S. Mirjalili).

https://doi.org/10.1016/j.asoc.2017.11.006
1568-4946/

Nature-inspired algorithms are mostly metaheuristics inspired from nature [14]. There are three main sources of inspiration: evolutionary-based (*e.g.* evolutionary algorithms and artificial immune systems), swarm-based (*e.g.* ant colony, bees colony, and particle swarm optimization), and physics-based–based (*e.g.* simulated annealing [13]). Two contradictory criteria that are common in all these techniques are: diversification (exploration of the search space) and intensification (exploitation of the best solutions found) [13].

As mentioned above, swarm-based algorithms mimic the collective behavior of natural creatures such as bat, grey wolf, ant lion, moth, etc. [15]. Recently, new evolutionary algorithms are proposed and shown a good performance when dealing with the Feature Selection (FS) problem. For instance, Ant Lion Optimizer (ALO) was utilized to tackle this problem in [16] and [17] as a wrapper feature selection model. In addition, a binary version of the ALO algorithm was presented in [18]. This approach has been enhanced by using a set of chaotic functions in adapting the single parameter that controls the balance between exploration and exploitation in the original algorithm [19]. Grey Wolf Optimizer (GWO) is a recent algorithm [20] that has been successfully employed for solving feature selection problems in [21,22]. Recently, a new wrapper-based feature selection algorithm that uses a hybrid Whale Optimization algorithm (WOA) with Simulated algorithm as a search method was proposed in [23].

Moreover, many researchers tried to implement stochastic methods to solve feature selection problems such as Ant Colony (AntRSAR), Genetic Algorithm (GenRSAR) by Jensen and Shen [24,25], Scatter Search (SSAR) by Jue et al. [26]. A chaos-based genetic feature selection optimization method (CGFSO) has been proposed in [27]. A hybrid approach has been proposed in [28] between GA and SA and between GA and Record to Record algorithm in [29]. The Particle Swarm Optimization algorithm (PSO) was used in feature selection approaches in [30–33]. It is worth mentioning here that some of the successful recent works on feature selection are done based on matrix computations. For example, the column-subset selection problems [34–36] are known to do feature selection with provable theoretical bounds. These methods have been used to do feature selection on k-means [37], Support Vector Machines [38], and Ridge Regression [39] which provide provable performance guarantees. These methods are known to outperform existing methods such as mutual information [40,41], recursive feature elimination [42–44], etc.

Due to the stochastic nature of the above-mentioned techniques, there is no guarantee to find the best set of features in feature selection problems. Also, the No-Free-Lunch (NFL) theorem for optimization asserts that there is no optimizer that is good enough to solve all optimization problems. Therefore, this means that the current stochastic-based feature selection methods may suffer from degraded performance when solving some problems. This motivated our attempts to investigate the efficiencies of the recently proposed the Whale Optimization Algorithm (WOA) [45] in the field of feature selection. WOA is an evolutionary algorithm that mimics the foraging behavior of the humpback whales in nature. To the best of our knowledge, no one has yet systematically applied this algorithm to feature selection problems [60].

In this paper, a wrapper feature selection approach is proposed based on WOA to find the minimal feature subsets. The proposed approach utilizes the main operators of WOA but modifies some of them in order to solve binary problems since the original version of this algorithm was created to solve continuous problems. The contributions of the paper are the proposal of the binary version of WOA and the integration of several evolutionary operators (selection, crossover and mutation) which have been used to improve both exploration and exploitation of this algorithm. The results showed that the involvement of the cross over and mutation operators in the WOA algorithm (WOA-CM) performs better than other approaches. The efficiency of the proposed WOA-CM algorithm is tested on several small, medium, and large size datasets, and compared with three nature-inspired algorithms which are GA, PSO, and ALO. In addition, WOA-CM is compared with five different filter-based feature selection methods. The conducted experiments show that WOA-C has a better performance compared to GA, PSO, and ALO. The adaptive mechanisms in the WOA-CM algorithm accelerate the convergence speed proportional to the number of iterations. They also balance exploration and exploitation efficiently to first avoid a large number of local solutions in feature selection problems and second to find an accurate estimation of the best solution.

The rest of this paper is organized as follows: Section 2 briefly introduces the WOA algorithm. Section 3 presents the details of the proposed approach. In section 4, the experimental results are presented and analysed. Finally, in Section 5, conclusions and future work are given.

2. Whale optimization algorithm

WOA belongs to the family of stochastic population-based algorithms proposed by Mirjalili and Lewis in 2016 [45]. It mimics the bubble-net feeding in the foraging behavior of the humpback whales [45]. The humpback whales hunt close to the surface with trapping the prey in a net of bubbles. They create this net when swimming in a '6'-shaped path.

The algorithm mimics two phases: the first phase (exploitation phase) is encircling a prey and spiral bubble-net attacking method, and the second phase (exploration phase) is searching randomly for a prey. The details of each phase are presented in the following subsections.

2.1. Exploitation phase (encircling prey/bubble-net attacking method)

To update a solution, Eqs. (1) and (2) are used, which mathematically model the movement of a whale around a prey [45].

$$\vec{D} = |\vec{C}.\vec{X^*}(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1) = \vec{X^*}(t) - \vec{A}.\vec{D} \tag{2}$$

where t represents the current iteration, X^* represents the best solution obtained so far, X is the current solution, $|\ |$ is the absolute value, and $\cdot$ is an element-by-element multiplication. A and C are coefficient vectors that are calculated as in Eqs. (3) and (4):

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

where a decreases linearly from 2 to 0 and r is a random vector in [0,1]. According to Eq. (2) the solutions update their positions according to the position of the best known solution. The adjustment of the values of A and C vectors control the areas where a solution can be located in the neighborhood of the best solution.

The humpback whales move in a shrinking encircling mechanism and along a spiral-shaped path towards the prey. In WOA, the Shrinking encircling behavior is simulated by decreasing the value of a in Eq. (3) according to Eq. (4).

$$a = 2 - t\frac{2}{MaxIter} \tag{5}$$

where t is the iteration number and $MaxIter$ is the maximum number of allowed iterations. The spiral-shaped path is achieved by calculating the distance between the solution (X) and the leading

solution (X^*). Then a spiral equation is created between the current solution and the best (leading) solution as in Eq. (6).

$$\vec{X}(t+1) = D'.e^{bl}.\cos(2\pi l) + \vec{X^*}(t) \quad (6)$$

where $\vec{D}$ is the distance between a whale X and a prey ($\vec{D} = |\vec{X^*}(t) - \vec{X}(t)|$), b defines the spiral's shape of the spiral, and l is a random number in [−1,1].

To model the two mechanisms; the shrinking encircling mechanism and the upward spiral-shaped path, a probability of 50% is assumed to choose between them during the optimization as en Eq. (7).

$$\vec{X}(t+1) = \begin{cases} ShrinkingEncircling(Eq.2) & if(p < 0.5) \\ spiral - shapedpath(Eq.6) & if(p \geq 0.5) \end{cases} \quad (7)$$

where p is a random number in [0,1]

2.2. Exploration phase (search for prey)

In order to enhance the exploration in WOA, instead of requiring the solutions to search randomly based on the position of the best solution found so far, a randomly chosen solution is used to update the position accordingly. So, a vector A with the random values greater than 1 or less than −1 is used to force a solution to move far away from the best known search agent. This mechanism can be mathematically modeled as in Eqs. (8) and (9).

$$\vec{D} = |\vec{C}.\vec{X_{rand}} - \vec{X}| \quad (8)$$

$$\vec{X}(t+1) = \vec{X_{rand}} - \vec{A}.\vec{D} \quad (9)$$

where $\vec{X_{rand}}$ is a random whale chosen from the current population.

The illustration of position updating of solutions in WOA using the main equations discussed above is given in Fig. 1. It may be seen that solutions are able to relocate around a solution in random locations or along a spiral path.[1]

[1] Note that the source codes of the WOA algorithm can be found at https://se.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm.

3. The proposed approach

In WOA, each feature subset can be seen as a position of a whale. Each subset may contain N features, where N is the number of features in the original set. The less the number of features in the solution and the higher the classification accuracy, the better is the solution. Each solution is evaluated according to the proposed fitness function, which depends two objectives: the solution's accuracy obtained by the KNN classifier and the number of selected features in the solution.

The algorithm starts with a set of randomly generated solutions (subsets) called population. Each solution is then evaluated using the proposed fitness function. The fittest solution in the population is marked as X^* (prey). The main loop in WOA is iterated a number of times. In each iteration, solutions update their positions according to each other to mimic the bubble-net attacking and ssearching for prey methods. To mimic the bubble-net attacking, a probability of 50% is assumed to choose between the shrinking encircling mechanism (Eq. (2)) and the spiral model (Eq. (9)) to update the position of the solutions (whales). When the shrinking encircling mechanism is employed, a balance between exploration and exploitation is required. A random vector A which contains values >1 or <1 is used for this purpose. If A > 1 then the exploration (searching for prey methods) is employed by searching in the neighborhood of a randomly selected solution, while the neighborhood of best solution so far is exploited when A < 1. This process is repeated until satisfying the stopping criteria, which is usually the maximum number of iterations. The pseudo codes of WOA are shown in Algorithm 1.

Algorithm 1 *(Pseudo-code of the WOA algorithm).*

```
Generate Initial Population Xi (i = 1, 2, ..., n)
Calculate the objective value of each solution
X*=the best solution
while (t < Max_Iteration)
    for each solution
        Update a, A, C, l, and p
        if1 (p<0.5)
            if2 (|A| < 1)
                Use Eq. (2) to update the position of the current solution
            else if2 (|A| >1)
                Select a random solution (Xrand)
                Use Eq. (9)
            end if2
        else if1 (p≥0.5)
            Update the position of the current search by the Eq. (6)
        end if1
    end for
    Check if any solution goes beyond the search space and amend it
    Calculate the fitness of each solution
    If there is a better solution, update X*
    t=t+1
end while
return X*
```

3.1. Solution representation

When designing a metaheuristic algorithm, representing the solution of the problem in hand is one of the main challenges. In this work, the solution is a one-dimensional vector that contains N elements, where N is the number of features in the original dataset. Each cell in the vector has a value of "1" or "0". Value "1" indicates that the corresponding feature is selected; otherwise, the value is set to "0".

3.2. Fitness function

The fitness function used in the proposed approach is designed to have a balance between the number of selected features in each solution (minimum) and the classification accuracy (maximum)

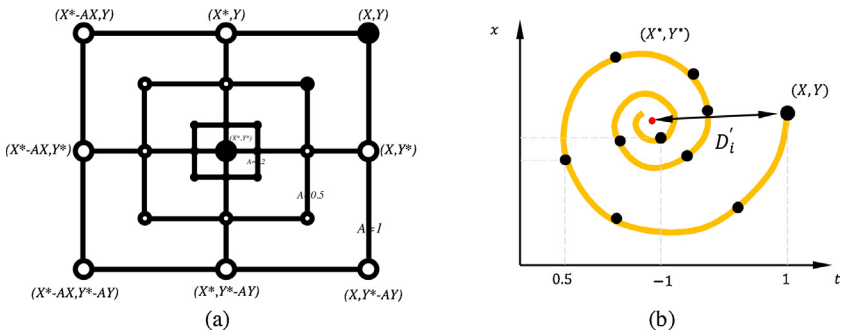


Fig. 1. Bubble-net search mechanism implemented in WOA (X* is the best (leading) solution obtained so far): (a) shrinking encircling mechanism (b) spiral updating position.

obtained by using these selected features, Eq. (10) represents the fitness function to evaluate solutions.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \tag{10}$$

where $\gamma_R(D)$ represents the classification error rate of a given classier (the K-nearest neighbor (KNN) classifier [46] is used here). $|R|$ is the cardinality of the selected subset and $|C|$ is the total number of features in the dataset, α and β are two parameters corresponding to the importance of classification quality and subset length. $\alpha \in [1,0]$ and $\beta = (1-\alpha)$ are adopted from [18].

This paper aims to study two aspects of WOA algorithm. For one, the influence of employing different selection mechanisms such as Tournament Selection [47] and the Roulette Wheel Selection [48] instead of the random selection is studied on the performance of the original WOA where two approaches are proposed; WOA-T (Whale Optimization Algorithm with tournament selection) and WOA-R (Whale Optimization Algorithm with roulette wheel selection). For another, the behavior of WOA is studied when using simple crossover and mutation operators to enhance the exploitation capability in WOA algorithm (WOA-CM).

In WOA, each whale (solution) changes its location according to Eq. (2) or Eq. (9). In the proposed approach (WOA-CM), however, the mutation operation is employed to simulate changing the position of a specific solution around a randomly selected solution (Eq. (2)) or around the best found solution (Eq. (9)). The mutation rate (r) used in this paper is given in Eq. (11). The parameter r is linearly decremented from 0.9 to 0 depending on the iteration number (i).

$$r = 0.9 + \frac{-0.9 * (i-1)}{MaxIteration - 1} \tag{11}$$

The crossover operation between the resultant solution from the mutation operation and the solution is employed as shown in Eq. (12).

$$X_i^{t+1} = Crossover(X^{Mut}, X_i^t) \tag{12}$$

where X^{Mut} is the resultant solution from the mutation operation, t is the current iteration. *Crossover (x,y)* is a simple operator that attempts to obtain an intermediate solution (Xd) between the solution x and y by switches between the two input solutions with the same probability as shown in Eq. (13).

$$X^d = \begin{cases} X_1 d & if(p \geq 0.5) \\ X_2 d & if(p < 0.5) \end{cases} \tag{13}$$

where d shows the *d-th* dimension, X1dis the first parent (reference solution one), and X2d shows the second parent (reference solution two).

After all, the pseudocode of the proposed algorithm is shown in Algorithm 2.

Algorithm 2 *(Pseudo-code of the WOA-CM approach).*

Generate Initial Population X_i *(i = 1, 2, ..., n)*
Calculate the objective value of each solution
X=the best solution*
while *(t < Max_Iteration)*
  for *each solution*
    Calculate the mutation rate (r) as in Eq. (24).
    Update a, A, C, l, and p
    if1 *(p <* 0.5*)*
      if2 *(|A| < 1)*
        Apply mutation operation on X (best solution) given mutation rate (r) to get* X^{Mut}
        Perform crossover between X^{Mut} *and* X_t *and set the new position of* X_t *to the output of crossover.*
      else if2 *(|A| >1)*
        Select a random search agent (X_{rand})
        Apply mutation operation on X_{rand} *given mutation rate (r) to get* X^{Mut}
        Perform crossover between X^{Mut} *and* X_t *and set the new position of* X_t *to the output of crossover.*
      end if2
    else if1 *(p ≥* 0.5*)*
      Use Eq. (6) to update the position of the current solution
    end if1
  end for
  Check if any solution goes beyond the search space and amend it
  Calculate the fitness of each solution
  *If there is a better solution, update X**
  t=t+1
end while
*return X**

It is worth mentioning here that the WOA is a non-metaphor algorithm since the inspiration is a valid natural behavior (spiral movement and bubble-net foraging), the mathematical model differs from those of existing algorithms and has been proposed from scratch, and the term "solution" is used to describe this algorithm. There is no metaphor (*e.g.* whale, water, ocean, age. etc.) to cover the novelty of the WOA algorithm and the preceding paragraphs presented the mathematical models in details. The mechanism of search might be more efficient on some problems compared to the well-known algorithm such as PSO and GA as discussed below:

In PSO, there are two vectors: position and velocity. The velocity vector is the main vector, which considers the best solution obtained so far by the particle and the entire swarm. With the velocity vector, the position vector is calculated for every solution. In WOA, however, there is only one vector to store the position of each solution in a search space. Also, the solutions are updated using a spiral equation (Eq. (6)) towards a promising solution or randomly in the search space using Eq. (2). The WOA is more efficient compared to PSO in terms of memory since this algorithm only stores the best estimation of the global optimum in each iteration. However, PSO stores the best solution and personal best for all particles.

GA selects solutions proportional to their fitness value and combines them to find better solutions. There is also a mutation to randomly modify some of the variables in the solutions. However, the solutions in WOA are changed based on one solution only. This solution might be randomly chosen or the best solution obtained so far. In GA, there is no mathematical equation and the solutions are combined with exchanging the variables, whereas the WOA algorithm requires solutions to change their elements using two mathematical equations: Eqs. (2) and (6).

WOA uses random solutions to lead the search more than PSO. Therefore, this algorithm shows a better exploratory behaviors compared to PSO. The problem of feature selection has a large number of local solutions, so WOA is potentially able to avoid those solutions better than PSO. In addition, WOA is equipped with adaptive mechanisms and is able to accelerate exploitation proportional to the number of iterations. This makes it more likely to obtain more accurate solutions in feature selection problems compared to GA with crossover operators that abruptly fluctuates the solutions throughout the optimization process.

4. Experimental results and discussions

4.1. Experimental setup

The implementation of the proposed algorithm is done using Matlab. Eighteen UCI [49] benchmark datasets were used to assess the efficiency of the proposed WOA based approaches. The datasets are presented in Table 1. A wrapper approach-based on the KNN classifier (where $K=5$ [18]) is used to generate the best reduct. In the proposed approach, each dataset is divided in cross validation in a same manner to that in [50] for evaluation. In K-fold cross-validation, $K-1$ folds are used for training and validation and the remaining fold is used for testing. This process is repeated M times. Hence, individual optimizer is evaluated K^*M times for individual data set. This size of training and validation sample sets is identical. The experiments were conducted on two phases. In the first phase, small and medium datasets were used in evaluating the proposed approaches, and in the second phase, two large datasets were used.

The experiments are tested on an Intel machine Core i5 CPU 2.2 GHz and 4 GB RAM and the parameter values of 10 and 100 have been utilized for the population size and maximum iteration respectively. Their parameters are selected after performing an extensive experimental study as will be explained in a later subsection

Table 1
List of datasets employed as cases studies in this work.

	Dataset	No. of Attributes	No. of Objects
1.	Breastcancer	9	699
2.	BreastEW	30	569
3.	CongressEW	16	435
4.	Exactly	13	1000
5.	Exactly2	13	1000
6.	HeartEW	13	270
7.	IonosphereEW	34	351
8.	KrvskpEW	36	3196
9.	Lymphography	18	148
10.	M-of-n	13	1000
11.	PenglungEW	325	73
12.	SonarEW	60	208
13.	SpectEW	22	267
14.	Tic-tac-toe	9	958
15.	Vote	16	300
16.	WaveformEW	40	5000
17.	WineEW	13	178
18.	Zoo	16	101
19.	Colon	2000	62
20.	Leukemia	7129	72

WOA is compared with state-of-the-art feature selection algorithms and other swarm-based optimization techniques using the following criteria:

- Classification accuracy: it is obtained by using the selected features on the test dataset. The average accuracy gained from 20 runs is calculated.
- Fitness values: they are obtained from each approach as reported. The mean, min and max fitness values are compared.
- Average selection size: it is the third comparison that has been presented in this section.

4.2. Parameters study of WOA

To study the impact of the standard parameters of the WOA, a set of extensive experiments has been performed using different values of the main parameters in the algorithm, (*i.e.*, number of search agents (n), maximum number of iterations (t), and the vector a in Eq. (5)). Three datasets were selected randomly (*i.e.*, one small dataset, one medium dataset, and one large dataset) to test the different combinations of these parameters. The number of search agents (n) parameter is allowed to take 6 different values: 5, 10, 20, 30, 40, or 50, and t parameter is allowed to take 5 different values: 20, 30, 50, 70 or 100, and the last parameter (vector a) is changed linearly in three different intervals: [0,1], [0,2], and [0,4].

A set of independent experiments are done for each dataset by varying the values of n, t and a simultaneously to illustrate the effect of these parameters on the performance of the WOA algorithm. A total of 90 parameter combinations were computed for each dataset. The algorithm was run 10 times for every set of parameter values for each dataset to be able to calculate the average accuracy and fitness value and reliably compare the results. Figs. 2–4 show how the performance of WOA relates to the choice of number of search agents n, when also varying the other parameters (i.e., t, and vector a) for small, medium, and large size datasets, respectively. The parameters study of the WOA on different types of data sets shows that 100 iterations were sufficient for the algorithm to obtain the best results in most of the cases as shown in Figs. 3–5. Moreover, a small number of agents between 10 and 20 was able to reach the highest accuracy for most of the benchmarks datasets.

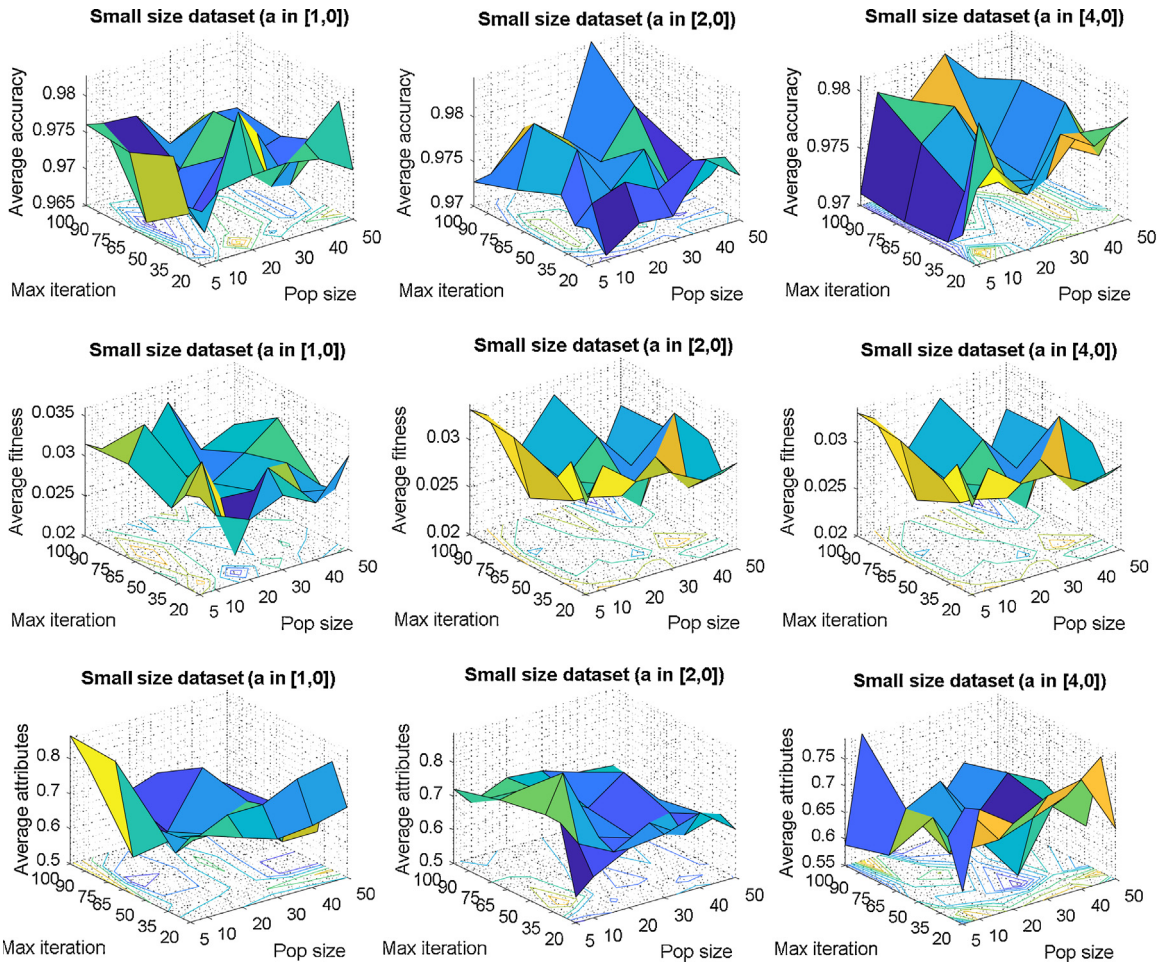


Fig. 2. Average accuracy, fitness and selection ratio for the small size dataset when changing the main controlling parameters of the proposed WOA-based method.

Table 2
Comparison between the proposed approaches with different selection mechanisms based on Selection Ration, Classification Accuracy and Computational Time.

Dataset	Average Selection Ratio			Classification Accuracy			Time		
	WOA	WOA-T	WOA-R	WOA	WOA-T	WOA-R	WOA	WOA-T	WOA-R
Breastcancer	**0.594**	0.661	0.628	0.9571	**0.9590**	0.9576	**1.360**	2.693	2.762
BreastEW	0.692	**0.685**	0.725	**0.9553**	0.9498	0.9507	**1.569**	3.023	3.247
CongressEW	0.647	0.641	**0.563**	**0.9296**	0.9147	0.9106	**1.149**	2.313	2.302
Exactly	0.831	0.827	**0.758**	0.7576	0.7396	**0.7633**	**2.346**	4.593	4.603
Exactly2	0.442	0.692	**0.204**	0.6985	**0.6994**	0.6907	**2.010**	4.338	4.040
HeartEW	0.665	0.646	**0.588**	0.7633	**0.7652**	0.7633	**0.955**	1.941	1.902
IonosphereEW	0.631	0.594	**0.550**	**0.8901**	0.8844	0.8801	**1.118**	2.175	2.273
KrvskpEW	0.775	**0.742**	0.768	**0.9151**	0.8965	0.9018	**31.200**	57.115	64.040
Lymphography	0.586	**0.519**	0.544	**0.7858**	0.7786	0.7595	**0.846**	1.715	1.696
M-of-n	**0.754**	0.812	0.796	0.8540	0.8389	**0.8603**	**2.229**	4.613	4.666
PenglungEW	0.444	0.472	**0.360**	0.7297	**0.7365**	0.7122	**1.126**	2.072	2.129
SonarEW	0.723	**0.637**	0.668	0.8543	**0.8611**	0.8572	**1.021**	1.978	2.039
SpectEW	0.550	0.523	**0.359**	0.7877	**0.7922**	0.7787	**0.939**	1.841	1.874
Tic-tac-toe	**0.739**	0.761	0.794	**0.7511**	0.7363	0.7398	**2.025**	4.036	4.514
Vote	0.463	0.513	**0.431**	**0.9387**	0.9350	0.9323	**0.985**	1.961	1.958
WaveformEW	**0.830**	0.843	0.856	**0.7127**	0.7101	0.7121	**86.421**	172.374	181.811
WineEW	**0.681**	0.685	0.719	**0.9281**	**0.9281**	0.9258	**0.862**	1.741	1.722
Zoo	**0.619**	0.731	0.747	**0.9647**	**0.9647**	0.9569	**0.859**	1.710	1.703

4.3. Results and discussion for small or medium-Sized datasets

To benchmark the performance of the proposed approaches, two methodologies were followed. The first one is conducting comparisons between the proposed approaches. These comparisons are done in Tables 2 and 3. The second one is comparing the best among the proposed approaches against three other optimizers from the literature (ALO, PSO, and GA adopted from [17]). These results are presented in Tables 4 and 5 where the results of ALO, PSO, and GA were obtained from [17].

The performance of the WOA, WOA-T and WOA-R over the two objectives (average selection size and classification accuracy) in addition to the computational time are outlined in Table 2. From the Table, it is evident that the use of a selection mecha-

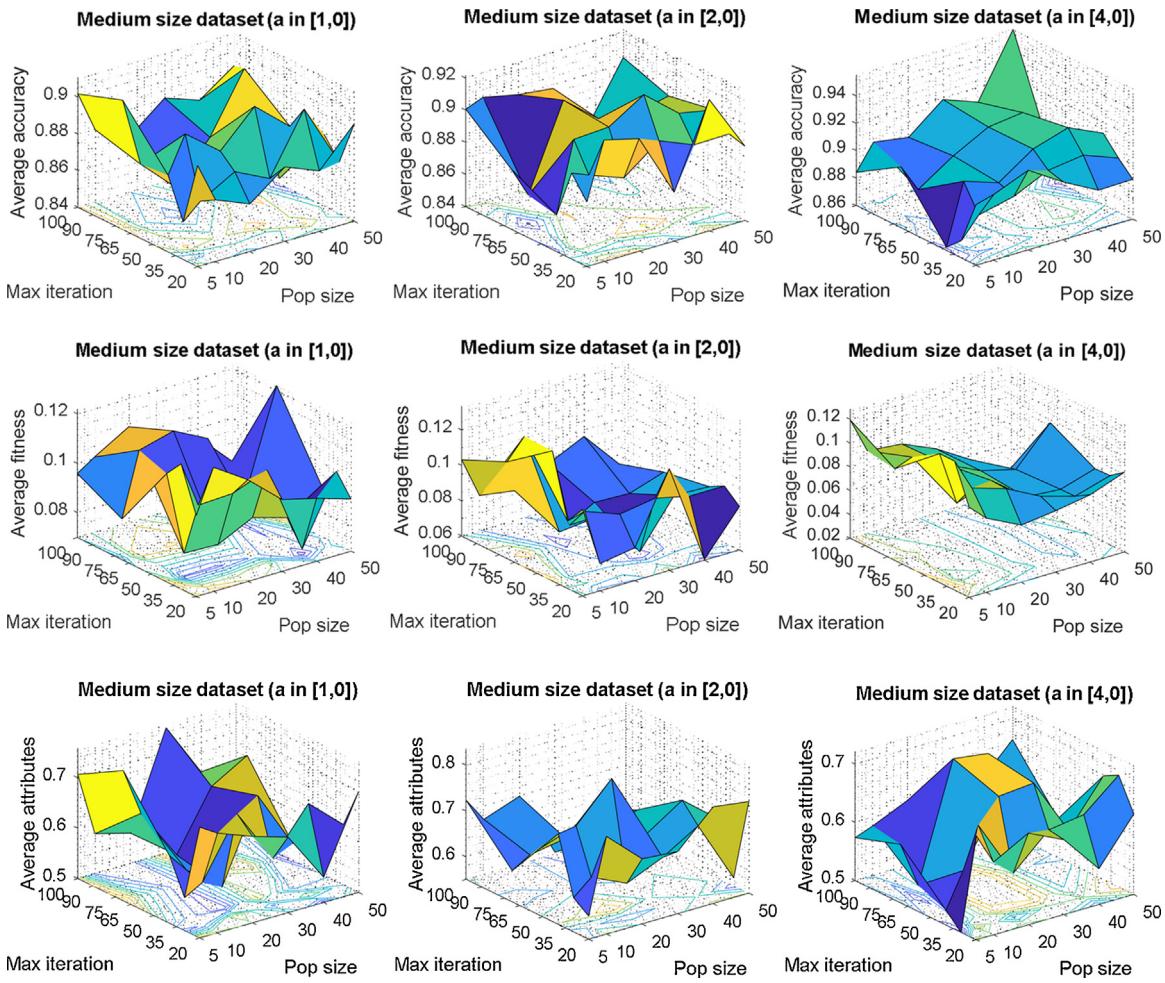


Fig. 3. Average accuracy, fitness and selection ratio for the medium size dataset when changing the main controlling parameters of the proposed WOA-based method.

Table 3
Comparison between the proposed approaches with simple Crossover and Mutation operators based on Selection Ration, Classification Accuracy and Computational Time.

Dataset	AVG Selection		Classification Accuracy		Time	
	WOA	WOA-CM	WOA	WOA-CM	WOA	WOA-CM
Breastcancer	0.594	**0.478**	0.9571	**0.9683**	1.3596	**1.3223**
BreastEW	0.692	**0.527**	0.9553	**0.9707**	1.5691	**1.5001**
CongressEW	0.647	**0.403**	0.9296	**0.9560**	**1.1485**	1.1796
Exactly	0.831	**0.465**	0.7576	**1.0000**	2.3459	**1.8662**
Exactly2	0.442	**0.404**	0.6985	**0.7421**	2.0100	**1.9627**
HeartEW	0.665	**0.535**	0.7633	**0.8067**	**0.9547**	0.9841
IonosphereEW	0.631	**0.424**	0.8901	**0.9256**	1.1180	**1.0782**
KrvskpEW	0.775	**0.515**	0.9151	**0.9718**	31.1998	**23.8776**
Lymphography	0.586	**0.456**	0.7858	**0.8518**	**0.8456**	0.8812
M-of-n	0.754	**0.462**	0.8540	**0.9914**	2.2292	**1.8585**
PenglungEW	0.444	**0.394**	0.7297	**0.7919**	1.1256	**0.9895**
SonarEW	0.723	**0.594**	0.8543	**0.9188**	1.0206	**0.9882**
SpectEW	0.550	**0.366**	0.7877	**0.8657**	**0.9391**	0.9665
Tic-tac-toe	**0.739**	0.767	0.7511	**0.7854**	2.0248	**1.9697**
Vote	**0.463**	**0.463**	**0.9387**	**0.9387**	**0.9855**	**0.9855**
WaveformEW	0.830	**0.635**	0.7127	**0.7533**	86.4208	**72.1484**
WineEW	0.681	**0.523**	0.9281	**0.9590**	**0.8617**	0.8989
Zoo	0.619	**0.375**	0.9647	**0.9804**	**0.8590**	0.8837

nism instead of the random selection mechanism has improved the performance of the proposed algorithm. In terms of the average selection ratio, WOA-R outperformed the native WOA algorithm in twelve datasets, while WOA-T outperformed WOA in nine datasets. This clearly shows the influence of using a selection mechanism instead of randomly selecting solutions. The classification accuracies reported in the same table support the previous discussion since the Tournament selection-based approach could outperform WOA on nine datasets as well. However, the time complexity of the native algorithm seems to be less than other approaches.

From the previous results we can see how the selection mechanism enhanced the performance of the WOA algorithm on some datasets, while, in the other datasets, the results of the native WOA algorithm are better than those of the enhanced approaches.

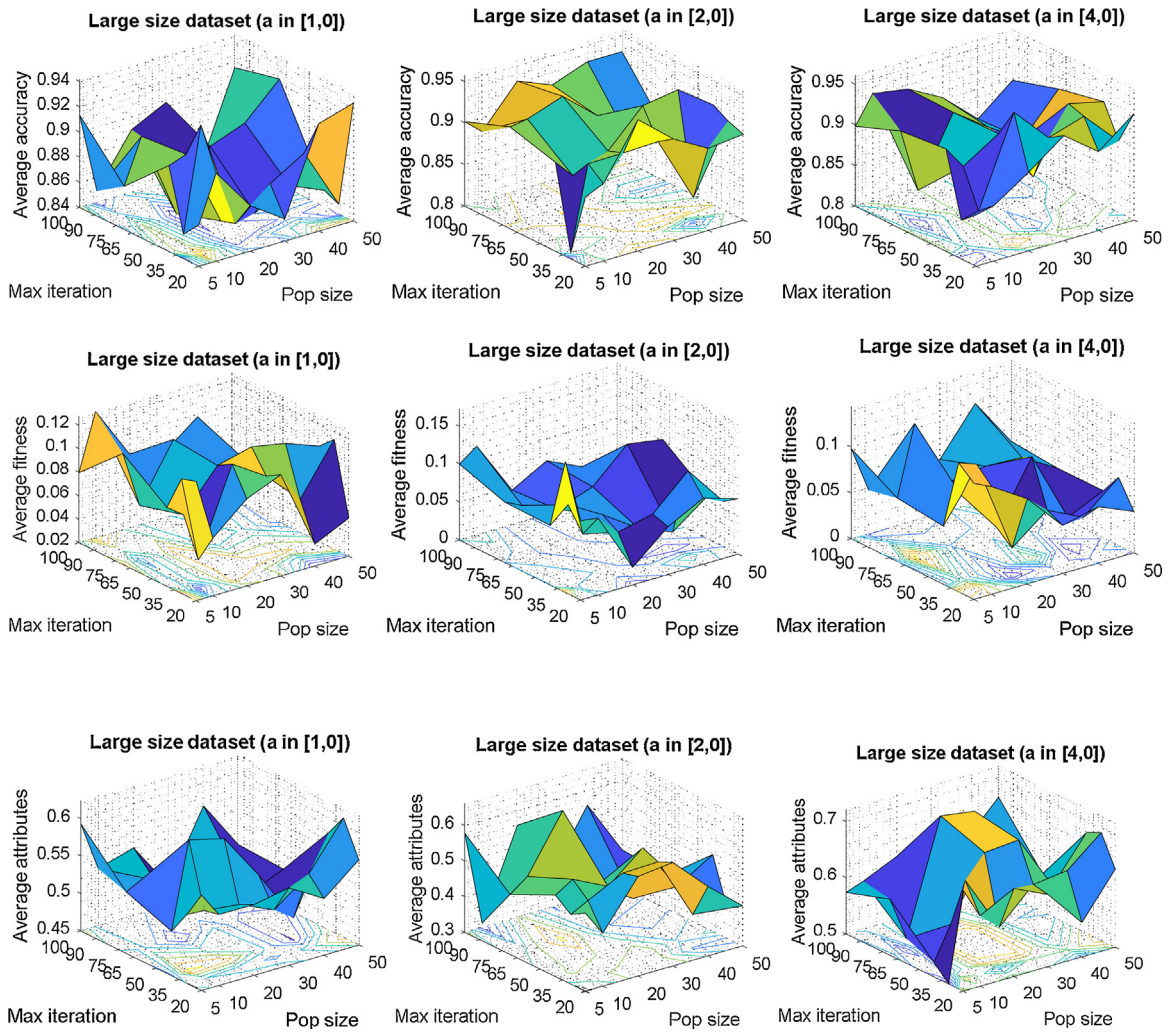


Fig. 4. Average accuracy, fitness and selection ratio for the large size dataset when changing the main controlling parameters of the proposed WOA-based method.

Table 4
Average selection Ratio of features from the different optimizers.

Dataset	Attributes	Instances	Selection Ration			
			WOA-CM	ALO	GA	PSO
Breastcancer	9	699	**0.478**	0.698	0.566	0.636
BreastEW	30	569	**0.527**	0.536	0.545	0.552
CongressEW	16	435	**0.403**	0.436	0.414	0.427
Exactly	13	1000	**0.465**	0.509	0.832	0.750
Exactly2	13	1000	**0.404**	0.823	0.475	0.475
HeartEW	13	270	**0.535**	0.793	0.730	0.611
IonosphereEW	34	351	0.424	**0.277**	0.509	0.564
KrvskpEW	36	3196	**0.515**	0.686	0.623	0.578
Lymphography	18	148	**0.456**	0.614	0.614	0.499
M-of-n	13	1000	**0.462**	0.852	0.525	0.695
PenglungEW	325	73	**0.394**	0.505	0.545	0.550
SonarEW	60	208	0.594	0.632	0.555	**0.520**
SpectEW	22	267	**0.366**	0.734	0.534	0.568
Tic-tac-toe	9	958	0.767	0.777	0.761	**0.734**
Vote	16	300	0.463	0.595	**0.414**	0.550
WaveformEW	40	5000	0.635	0.893	0.632	**0.568**
WineEW	13	178	**0.523**	0.823	0.664	0.643
Zoo	16	101	**0.375**	0.873	0.632	0.609

This proves the capability of the native WOA algorithm to balance between exploration and exploitation. The native WOA randomly selects a search agent to update the positions of other search agents accordingly using Eq. (9). This proves a high exploration ability of the WOA in almost half of the iterations. However, the exploitation ability which is achieved by Eq. (7), that allows the search agents to move towards the best search agent obtained so far, is proved by these results. In addition, for nearly half of the datasets, the performance of WOA algorithm is improved due to the use of different selections mechanisms. A possible reason is that poor solutions are

Table 5
Average classification accuracy using selected feature by different feature selection methods on the test data.

Dataset	WOA-CM	ALO	GA	PSO	Full
Breastcancer	**0.968**	0.961	0.955	0.954	0.944
BreastEW	**0.971**	0.930	0.938	0.941	0.963
CongressEW	**0.956**	0.929	0.938	0.937	0.917
Exactly	**1.000**	0.660	0.666	0.684	0.673
Exactly2	0.742	0.745	**0.757**	0.746	0.743
HeartEW	0.807	**0.826**	0.822	0.784	0.815
IonosphereEW	**0.926**	0.866	0.834	0.843	0.866
KrvskpEW	**0.972**	0.956	0.923	0.942	0.915
Lymphography	**0.852**	0.787	0.708	0.692	0.683
M-of-n	**0.991**	0.864	0.927	0.864	0.849
PenglungEW	0.792	0.627	0.696	0.720	**0.951**
SonarEW	**0.919**	0.738	0.726	0.740	0.62
SpectEW	**0.866**	0.801	0.775	0.769	0.831
Tic-tac-toe	**0.785**	0.725	0.713	0.728	0.715
Vote	**0.939**	0.917	0.894	0.894	0.877
WaveformEW	0.753	**0.773**	0.767	0.761	0.768
WineEW	**0.959**	0.911	0.933	0.950	0.932
Zoo	**0.980**	0.909	0.884	0.834	0.792

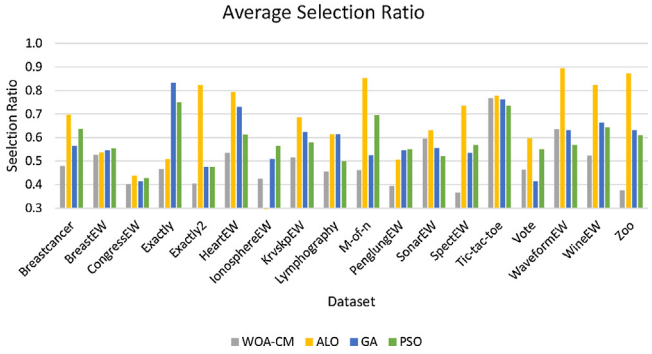


Fig. 5. Average Ratio of Selected Features for the Different Optimizers.

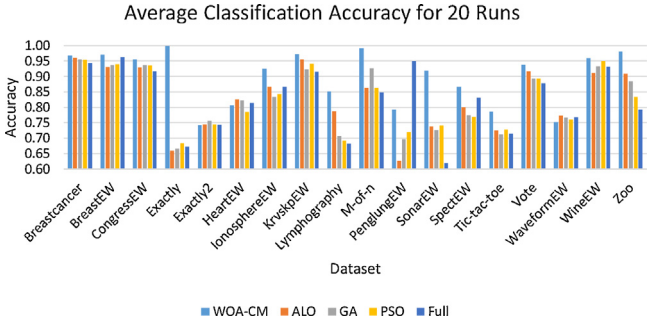


Fig. 6. Average Classification Accuracy for Different Feature Selection Algorithms.

given a chance to be selected in the exploration phase by roulette wheel and tournament selection mechanisms.

Furthermore, the results of WOA-CM and the native WOA algorithm are summarized in Table 3 based on the three criteria mentioned previously. The enhanced approach shows a very good performance over all tested datasets. In terms of minimal reducts and classification accuracy, WOA-CM outperformed WOA on all datasets except on one datasets where the two approaches obtained the same results. These results show that a good balance between the exploration and exploitation is necessary when using global search algorithms as WOA. The mutation operator has proved its capability to improve the performance of WOA algorithm. The computational time required by the two approaches is shown in the table. WOA-CM clearly performs better than the native approach and could obtain better results in a shorter time.

The results in Table 2 indicate how the selection mechanism is able to enhance the exploration ability of WOA algorithm. Correspondingly, Table 3 shows that using simple crossover and mutation operators improves the performance of the native algorithm in most of the all datasets in terms of classification accuracy, selected features ration, and computational time. This can be interpreted easily by saying that the mutation operator employed here plays the role of a local search which accordingly improves the exploitation ability of WOA algorithm. The results of WOA-CM are much better than that of the native approach in all evaluation criteria. This originates from the crossover and mutation operators that reduced the required computation time in the WOA-CM approach.

Since we obtained the results from the literature, the comparison of CPU time between algorithms is not possible. In addition, the run time of algorithms depends highly on the implementation and the programming language employed. In order to provide a fair comparison, the algorithms are compared based on their computational complexity. The computational complexity of WOA-CM is of $O(t(n^*p + Cof^*p))$ where t shows the number of iterations, n is the number of variables, p is the number of solutions, and cof indicates the cost of objective function. This computation complexity is equal to that of PSO. However, the computational complexity of GA and ALO is of $O(t^* (n^*p + Cof^*p + p^*log(p)))$ in the best case and $O(t^* (n^*p + Cof^*p + p^2))$ in the worst case. This shows that the computational complexity of GA and ALO is worse than those of WOA-CM and PSO due to the need to sort the solutions in each iteration.

From Table 4, one can observe the good performance of the WOA-CM approach compared to other approaches. It outperformed ALO algorithm on all datasets except on the IonosphereEW dataset. At the same time, our approach outperformed GA and PSO algorithms on fifteen datasets. Fig. 2 illustrates the performance of the four approaches.

The classification accuracy produced when using full datasets and using the selected features from all feature selection approaches is presented in Table 5. Note that the best results are highlighted in bold. The results of algorithms that provide discrepancies greater than or equal to a p-value of 0.05 compared to the best algorithms are underlined as well. From the table, one can observe that the worst classification accuracy is produced when using all features in a dataset. Moreover, it can be seen from the table that the proposed approach (WOA-CM) outperformed other approaches on most datasets (fourteen out of eighteen datasets). This result proves the ability of the proposed approach to explore the search space efficiently and find the best reducts that produce higher classification accuracy. The performance of all approaches is represented in Fig. 3.

Based on the results reported in Tables 4 and 5, and Fig. 6. it can be seen that WOA-CM provided the highest classification accuracy

Table 6
Mean, best and worst fitness values obtained from the different optimizers.

Dataset	Mean				Best				Worst			
	WOA-CM	ALO	GA	PSO	WOA-CM	ALO	GA	PSO	WOA-CM	ALO	GA	PSO
Breastcancer	0.035	**0.021**	0.028	0.03	0.034	**0.017**	**0.017**	0.026	**0.039**	0.026	0.039	0.034
BreastEW	0.034	**0.033**	0.036	**0.03**	0.026	0.032	**0.021**	**0.021**	0.041	0.037	0.047	**0.053**
CongressEW	0.047	0.046	0.043	**0.04**	0.039	0.034	**0.028**	0.034	0.061	**0.062**	0.055	0.041
Exactly	**0.005**	0.289	0.281	0.28	**0.005**	0.284	0.269	0.21	0.005	0.293	0.311	**0.323**
Exactly2	0.259	**0.24**	0.25	0.25	0.244	0.234	**0.216**	**0.216**	0.288	0.246	0.299	**0.305**
HeartEW	0.193	**0.122**	0.138	0.15	0.159	**0.111**	0.122	0.133	**0.220**	0.133	0.144	0.178
IonosphereEW	**0.076**	0.108	0.125	0.14	**0.037**	0.103	0.085	0.12	0.096	0.12	0.162	**0.171**
KrvskpEW	**0.027**	0.05	0.068	0.05	**0.022**	0.031	0.034	0.033	0.034	0.071	**0.126**	0.07
Lymphography	0.148	**0.136**	0.171	0.19	0.137	**0.082**	0.122	0.143	0.178	0.163	**0.265**	**0.265**
M-of-n	**0.005**	0.107	0.075	0.11	**0.005**	0.09	0.021	0.06	0.005	0.117	0.153	**0.159**
PenglungEW	0.203	**0.139**	0.22	0.22	0.165	**0**	0.125	0.125	0.218	0.208	**0.292**	**0.292**
SonarEW	**0.079**	0.179	0.13	0.13	**0.044**	0.13	0.072	0.072	0.111	**0.261**	0.232	0.217
SpectEW	0.135	**0.124**	0.137	0.13	0.114	**0.09**	0.124	0.101	**0.176**	0.146	0.146	0.157
Tic-tac-toe	**0.220**	0.222	0.242	0.24	0.217	**0.203**	0.209	0.209	0.237	0.237	0.259	**0.266**
Vote	0.050	**0.037**	0.054	0.05	**0.030**	0.03	0.03	0.03	0.072	0.05	**0.08**	**0.08**
WaveformEW	0.250	0.206	**0.203**	0.22	0.234	0.192	**0.189**	0.205	**0.272**	0.219	0.213	0.229
WineEW	0.045	0.017	**0.014**	0.02	0.038	**0**	**0**	**0**	**0.057**	0.034	0.034	0.034
Zoo	**0.023**	0.073	0.082	0.1	0.022	0.036	**0**	0.029	0.025	0.118	0.176	**0.206**
Total	**1.834**	2.148	2.299	2.39	**1.569**	1.698	1.685	1.767	2.137	2.541	3.034	**3.081**

and the minimum number of selected features in comparison with the other feature selection methods on 80% of the datasets. This is due to the fact that the global search property of WOA-CM originates from WOA and the simple crossover and mutation operators result in selecting the most informative features and eliminating the irrelevant and redundant features.

The results when solving all data sets are presented in Table 6. Here WOA-CM approach is used in comparison with other approaches. From the table, we can see that WOA-CM outperforms ALO, PSO and GA in Mean and Best Fitness criterion. At the same time, it is not worse than any other approach on all dataset. The difference between WOA-CM and other approaches is statistically significant. By performing a *t*-test (with $p < 0.05$) on the classification accuracy, WOA-CM algorithm significantly differ from ALO and PSO with $p = 0.010358$ and $p = 0.004505$, respectively. In addition, the *t*-test shows that WOA-CM is significantly better than GA with $p = 0.007313$. According to these results, we can conclude that WOA-CM provides significantly better results compared to other approaches at a significance level of 0.05.

This better performance of the WOA algorithm proves its ability to balance the exploration and exploitation during the optimization process. Based on the results obtained, the performance of the whale optimizer is proved on the large datasets as well as small size data sets. The three datasets; penglungEW, krvskpEW and ionosphereEW are relatively large datasets and the fitness value of the proposed approach is obviously less than all other approaches. Inspecting the best and worst columns in the table of results, it is evident that WOA-CM outperforms ALO, PSO and GA.

4.4. Comparison of WOA-CM and filter-based methods

To evaluate the performance of the proposed approach, its results were compared with the standard filter-based feature selection methods that were reported in [51]. To make a fair comparison, we selected five methods from two different categories; univariate filter-based methods (Information gain(IG) [52], Spectrum [53], Fisher Score (F-Score) [54]) where the dependencies between the features are not considered in the evaluation criterion, and multivariate filter methods (*e.g.* Correlation-based feature selection (CFS) [55], Fast Correlation Based Filter (FCBF) [56]) where the dependencies between features is considered in the evaluation of the relevance of features. These methods also attracted our attentions because they differ from each other in terms of using class labels of the training patterns to compute the relevance of each feature. The supervised methods (*e.g.* CFS, FCBF, F-Score and IG) use class labels while the unsupervised methods (*e.g.* Spectrum) do not employ any class labels to evaluate the feature relevancy (Fig. 6).

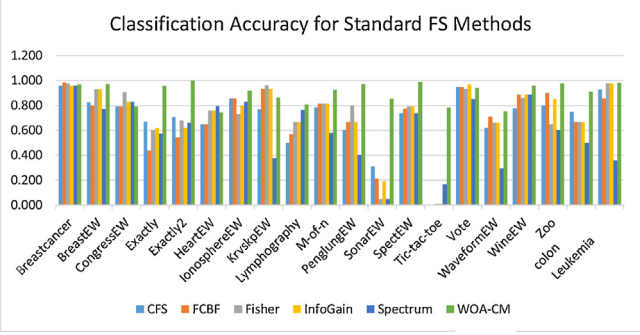


Fig. 7. Classification accuracy produced by using selected features from standard feature selection algorithms.

The classification accuracy obtained when using the selected features using the filter-based feature selection methods and WOA-CM approach in the 20 independent runs is presented in Table 7. Several experiments were conducted using several number of features, where the top 5, 10, and 20 features were used to produce the classification accuracy. For the PenglungEW, Colon and Lukemia datasets, it has been found that using the top ranked 20 features performed well, while, for the other datasets the top ranked 5 features performed well. Please note that the best results are highlighted in bold and underlined and the second best results are shown in boldface only. It can be seen from the table and Fig. 7 that the proposed approach (WOA-CM) outperformed other approaches on most datasets (fourteen out of eighteen datasets). It outperformed the supervised univariate feature selection methods (*e.g.*, F-Score and IG), the supervised multivariate feature selection method (*e.g.*, CFS, FCBF), and the unsupervised method (*i.e.* Spectrum). These results prove the ability of the proposed approach to explore the search space efficiently and find the best reducts that produce higher classification accuracy. Moreover, the obtained results prove the fact that the wrapper feature selection methods perform better than the filter-based techniques for classification tasks since both class labels and feature dependencies are considered in the selection of relevant feature subsets.

Table 7
Average classification accuracy using selected feature by standard feature selection methods and WOA-CM.

Dataset	CFS	FCBF	F-Score	IG	Spectrum	WOA-CM
Breastcancer	0.957	**0.986**	**0.979**	0.957	0.957	0.968
BreastEW	0.825	0.798	**0.930**	**0.930**	0.772	**0.971**
CongressEW	0.793	0.793	**0.908**	**0.828**	**0.828**	0.792
Exactly	**0.670**	0.440	0.600	0.615	0.575	**0.956**
Exactly2	**0.705**	0.545	0.680	0.620	0.660	**1.000**
HeartEW	0.648	0.648	**0.759**	**0.759**	**0.796**	0.742
IonosphereEW	**0.857**	**0.857**	0.729	0.800	0.829	**0.919**
KrvskpEW	0.768	**0.934**	**0.959**	**0.934**	0.377	0.866
Lymphography	0.500	0.567	0.667	0.667	**0.767**	**0.807**
M-of-n	0.785	**0.815**	**0.815**	**0.815**	0.580	**0.926**
PenglungEW	0.600	0.667	**0.800**	0.667	0.400	**0.972**
SonarEW	**0.310**	0.214	0.048	0.191	0.048	**0.852**
SpectEW	0.736	0.774	**0.793**	**0.793**	0.736	**0.991**
Tic-tac-toe	0.000	0.000	0.010	0.010	**0.167**	**0.785**
Vote	**0.950**	**0.950**	0.933	**0.967**	0.850	0.939
WaveformEW	0.620	**0.710**	0.662	0.662	0.292	**0.753**
WineEW	0.778	**0.889**	0.861	**0.889**	**0.889**	**0.959**
Zoo	0.800	**0.900**	0.650	0.850	0.600	**0.980**
Colon	**0.750**	0.667	0.667	0.667	0.500	**0.909**
Leukemia	0.929	0.857	**0.980**	**0.980**	0.357	**0.982**

Table 8
Results for high dimensional datasets.

	Accuracy	STDEV	Fitness			Time	Selection Ratio
			AVG	Min	Max		
Colon Dataset							
WOA	0.884	0.021	0.102	**0.04**	0.124	8.684	**0.006**
WOA-T	0.883	**0.010**	0.120	0.102	0.143	14.66	0.028
WOA-R	0.868	0.029	0.096	0.060	0.107	16.143	0.027
WOA-CM	**0.909**	0.017	**0.088**	0.044	**0.103**	**5.333**	0.026
ALO	0.866	0.025	0.179	0.13	0.261	20.07	0.056
GA	0.834	0.301	0.13	0.072	0.232	43.45	0.084
PSO	0.843	0.256	0.13	0.072	0.217	37.50	0.075
Leukemia Dataset							
WOA	0.963	0.021	0.026	0.018	0.042	43.984	0.036
WOA-T	0.965	0.012	0.038	0.022	0.056	67.778	0.096
WOA-R	0.951	0.016	0.026	0.017	0.042	73.663	**0.029**
WOA-CM	**0.982**	**0.007**	**0.02**	**0.005**	**0.022**	**20.42**	0.033
ALO	0.909	0.021	0.05	0.031	0.071	68.99	0.071
GA	0.884	0.017	0.068	0.034	0.126	180.23	0.102
PSO	0.834	0.019	0.05	0.033	0.07	102.33	0.097

4.5. Experimental results for high-dimensional datasets

To further prove the merits of the proposed algorithms in solving challenging feature selection problems, this sub-section employs two gene expression datasets: the colon cancer dataset and the leukemia dataset. The colon cancer consists of 2000 features and 62 instances. The goal is to discriminate between cancerous and normal tissues in a colon cancer problem [57]. The dataset was randomly split into a training set of 50 samples and a test set of 12 [58]. On the other hand, the leukemia dataset contains 7129 features and 72 instances. The goal is to distinguish between two classes of leukemia ALL and AML [59]. The first 38 samples are used as a training set and the remaining 34 as a test set [58]. Table 8 shows a summary of the results obtained for these datasets.

Inspecting the results in Table 8, it may be observed that WOA-CM performs better than other WOA approaches in terms of classification accuracy and the number of selected attributes on the Leukemia dataset. In addition, it is better than other approaches on the Colon dataset in terms of classification accuracy and it shows competitive results in terms of the selected attributes. In addition, WOA-CM outperformed other nature inspired FS algorithms (ALO, GA ad PSO) in selecting the minimal feature subsets that increased the classification accuracy with the minimum time cost.

In regard to the computation time, the leukemia dataset, which is the largest dataset employed in this work, was solved within only five seconds by WOA-CM as presented in Table 8. This table shows that the speed of the proposed algorithm was fast when solving the Colon cancer dataset as well.

Taken together, the WOA-CM outperformed GA due to a better exploitation. In a binary search space, the range of variables are limited to 0 and 1. Therefore, the optimization is different from a continuous search space where search agents of an algorithm can move around. A very high exploration results into changing the variables from 0 to 1 and vice versa. Although this is essential to avoid local optima, an algorithm needs to change the variables of a binary problem less frequently proportional to the number of iteration. In WOA-CM, the parameter a (Eq. (5)) smoothly enhances the exploitation of this algorithm and is the main mechanism for providing better results compared to GA. Moreover, WOA-CM shows superior performance compared to ALO and PSO. The adaptive mechanisms in the WOA-CM algorithms accelerate the convergence speed proportional to the number of iterations. They also balance exploration and exploitation efficiently to first avoid a large number of local solutions in feature selection problems, and to find an accurate estimation of the best solution.

5. Conclusion

Variants of WOA algorithm were studied and developed in this work. To the best of our knowledge, this was the first systematic

attempt to solve feature selection problems using the WOA algorithm. The continuous version of WOA was converted into binary by simply inspiring the basic operators of the WOA and replacing them with binary ones. Three improvements on the native WOA algorithm were proposed here WOA-T, WOA-R and WOA-CM. The proposed approaches were applied on the feature selection domain. Eighteen well-known datasets from UCI repository were used to assess the performance of these approaches. The results were compared to three other optimizers from the literature: GA, PSO, ALO, and five standard filter feature selection methods. Experimental results showed that our approaches are able to produce better results than the other approaches adopted from the literature on most of the datasets. We can see that when we used a simple mutation operator to enhance the local search (exploitation) in WOA algorithm, the performance of the algorithm becomes better than the native one where the algorithm becomes more efficient in searching the optimal/near optimal subsets in lower time rates.

As a future work, WOA algorithm may be proposed as a filter feature selection approach seeking to evaluate the generality of the selected features and study the classification accuracy by using many classifiers other than KNN which was adopted in this paper. Investigating the performance of WOA-CM algorithm will be a valuable contribution when applied to much higher dimensional datasets. In addition, a study on the effect of high dimensionality and small-sized samples versus large-sized samples with respect to the proposed method is an interesting work for the future.

Acknowledgements

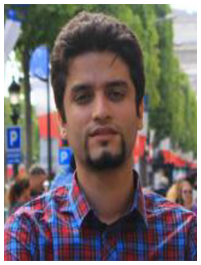
The authors would like to thank Dr. Hossam Faris, Dr. Ibrahim Aljarah, Dr. Abdelaziz I. Hammouri, and Mrs. Fatima Mafarjeh for their help in preparing the revised draft.

References

[1] R. Jensen, Combining rough and fuzzy sets for feature selection, in: School of Informatics, University of Edinburgh, 2005.
[2] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Boston, 1998.
[3] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[4] Z. Zhu, Y.S. Ong, M. Dash, Wrapper–filter feature selection algorithm using a memetic framework, IEEE Transactions on Systems, Man, and Cybernetics 37 (1) (2007) 70–76.
[5] M. Mafarja, S. Abdullah, Fuzzy Modified Great Deluge Algorithm for Attribute Reduction, in Recent Advances on Soft Computing and Data Mining, Springer, 2014, pp. 195–203.
[6] M. Mafarja, S. Abdullah, A fuzzy record-to-record travel algorithm for solving rough set attribute reduction, Int. J. Syst. Sci. 46 (3) (2015) 503–512.
[7] M. Mafarja, S. Abdullah, Modified great deluge for attribute reduction in rough set theory, in Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference On. 2011, IEEE (2011) 1464–1469.
[8] M. Mafarja, S. Abdullah, Record-to-record travel algorithm for attribute reduction in rough set theory, J. Theor. Appl. Inf. Technol 49 (2) (2013) 507–513.
[9] H. Abusamra, A comparative study of feature selection and classification methods for gene expression data of glioma, Procedia Comput. Sci. 23 (2013) 5–14.
[10] N. Zhong, J. Dong, S. Ohsuga, Using rough sets with heuristics for feature selection, J. Intell. Inf. Syst. 16 (3) (2001) 199–214.
[11] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (March) (2003) 1157–1182.
[12] C. Lai, M.J. Reinders, L. Wessels, Random subspace method for multivariate feature selection, Pattern Recognit. Lett. 27 (10) (2006) 1067–1076.
[13] E.G. Talbi, Metaheuristics From Design to Implementation, Wiley Online Library, 2009.
[14] I. Fister Jr., et al., A Brief Review of Nature-inspired Algorithms for Optimization, arXiv preprint arXiv:1307.4186, 2013.
[15] F. Valdez, in: J. Kacprzyk, W. Pedrycz (Eds.), Bio-Inspired Optimization Methods, in Springer Handbook of Computational Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 1533–1538.
[16] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[17] H.M. Zawbaa, E. Emary, B. Parv, Feature selection based on antlion optimization algorithm, 2015 Third World Conference on Complex Systems (WCCS) (2015) 1–7.
[18] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing (2016).
[19] H.M. Zawbaa, E. Emary, C. Grosan, Feature selection via chaotic antlion optimization, PLoS One 11 (3) (2016) e0150652.
[20] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[21] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[22] E. Emary, et al., Feature subset selection approach by gray-wolf optimization, in: Afro-European Conference for Industrial Advancement, Springer, 2015.
[23] M.M. Mafarja, S. Mirjalili, Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, Neurocomputing (2017).
[24] R. Jensen, Q. Shen, Finding rough set reducts with ant colony optimization, Proceedings of the 2003 UK Workshop on Computational Intelligence (2003) 15–22.
[25] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, IEEE Trans. Knowl. Data Eng. 16 (12) (2004) 1457–1471.
[26] W. Jue, A.R. Hedar, W. Shouyang, Scatter search for rough set attribute reduction, Bio-Inspired Computing: Theories and Applications, 2007. BIC-TA 2007. Second International Conference on (2007).
[27] H. Chen, et al., A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm, Math. Probl. Eng. (2013) 2013.
[28] M. Mafarja, S. Abdullah, Investigating memetic algorithm in solving rough set attribute reduction, Int. J. Comput. Appl. Technol. 48 (3) (2013) 195–202.
[29] M. Majdi, S. Abdullah, N.S. Jaddi, Fuzzy population-based meta-heuristic approaches for attribute reduction in rough set theory, World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng. 9 (12) (2015) 2289–2297.
[30] X. Wang, et al., Feature selection based on rough sets and particle swarm optimization, Pattern Recognit. Lett. 28 (4) (2007) 459–471.
[31] B. Chakraborty, Feature subset selection by particle swarm optimization with fuzzy fitness function, in: Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on, IEEE, 2008.
[32] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[33] R. Bello, et al., Two-step particle swarm optimization to solve the feature selection problem, in: Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications, IEEE Computer Society: Brazil, 2007, pp. 691–696.
[34] S. Paul, M. Magdon-Ismail, P. Drineas, Column selection via adaptive sampling, Advances in Neural Information Processing Systems (2015).
[35] C. Boutsidis, P. Drineas, M. Magdon-Ismail, Near-optimal column-based matrix reconstruction, SIAM J. Comput. 43 (2) (2014) 687–717.
[36] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Relative-error CUR matrix decompositions, SIAM J. Matrix Anal. Appl. 30 (2) (2008) 844–881.
[37] C. Boutsidis, P. Drineas, M.W. Mahoney, Unsupervised feature selection for the $ k $-means clustering problem, Advances in Neural Information Processing Systems (2009).
[38] S. Paul, M. Magdon-Ismail, P. Drineas, Feature selection for linear SVM with provable guarantees, AISTATS (2015).
[39] S. Paul, P. Drineas, Feature selection for ridge regression with provable guarantees, Neural Comput. (2016).
[40] P.A. Estévez, et al., Normalized mutual information feature selection, IEEE Trans. Neural Netw. 20 (2) (2009) 189–201.
[41] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency: max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[42] P.M. Granitto, et al., Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products, Chemom. Intell. Lab. Syst. 83 (2) (2006) 83–90.
[43] E. Gysels, P. Renevey, P. Celka, SVM-based recursive feature elimination to compare phase synchronization computed from broadband and narrowband EEG signals in brain–computer interfaces, Signal Process. 85 (11) (2005) 2178–2189.
[44] X. Zhang, et al., Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data, BMC Bioinf. 7 (1) (2006) 197.
[45] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[46] L.-Y. Chuang, et al., Improved binary PSO for feature selection using gene expression data, Comput. Biol. Chem. 32 (1) (2008) 29–38.
[47] B.L. Miller, D.E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, Complex Syst. 9 (3) (1995) 193–212.
[48] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford university press, 1996.
[49] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, 1998 ([cited 2016 1 June]; Available from:) http://www.ics.uci.edu/?mlearn/.
[50] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning Springer Series in Statistics, vol. 1, Springer, Berlin, 2001.

[51] Z. Zhao, et al., Advancing feature selection research, in: ASU Feature Selection Repository, 2010, pp. 1–28.
[52] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley & Sons, 2012.
[53] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007.
[54] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.
[55] M.A. Hall, L.A. Smith, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, FLAIRS Conference (1999).
[56] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, Proceedings of the 20th International Conference on Machine Learning (ICML-03) (2003).
[57] U. Alon, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. 96 (12) (1999) 6745–6750.
[58] L. Ke, Z. Feng, Z. Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recognit. Lett. 29 (9) (2008) 1351–1357.
[59] T.R. Golub, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[60] Sharawi Marwa, Hossam M. Zawbaa, E. Emary, Feature selection approach based on whale optimization algorithm, in: Advanced Computational Intelligence (ICACI), 2017 Ninth International Conference on, IEEE, 2017.

Majdi Mafarja received his B.Sc in Software Engineering and M.Sc in Computer Information Systems from Philadelphia University and The Arab Academy for Banking and Financial Sciences, Jordan in 2005 and 2007 respectively. He did his PhD in Computer Science at National University of Malaysia (UKM). He was a member in Datamining and Optimization Research Group (DMO). Now he is an assistant professor at the Department of Computer Science at Birzeit University. His research interests include Evolutionary Computation, Meta-heuristics and Data mining.



Seyedali Mirjalili is a lecturer in Griffith College, Griffith University. He received his B.Sc. degree in Computer Engineering (software) from Yazd University, M.Sc. degree in Computer Science from Universiti Teknologi Malaysia (UTM), and Ph. D. in Computer Science from Griffith University. He was a member of Soft Computing Research Group (SCRG) at UTM. His research interests include Robust Optimization, Multi-objective Optimization, Swarm Intelligence, Evolutionary Algorithms, and Artificial Neural Networks. He is working on the application of multi-objective and robust *meta*-heuristic optimization techniques in Computational.