Swarm and Evolutionary Computation 42 (2018) 29–42

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach

Hossam M. Zawbaa [a,b,*], E. Emary [c,d], Crina Grosan [a,e], Vaclav Snasel [f]

[a] Faculty of Mathematics and Computer Science, Babes-Bolyai University, Romania
[b] Faculty of Computers and Information, Beni-Suef University, Egypt
[c] Faculty of Computers and Information, Cairo University, Egypt
[d] Faculty of Computer Studies, Arab Open University, Cairo, Egypt
[e] College of Engineering, Design and Physical Sciences, Brunel University, United Kingdom
[f] Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Czech Republic

ARTICLE INFO

Keywords:
Bio-inspired optimization
Antlion optimization
Grey wolf optimization
Hybrid ALO-GWO
Swarm optimization
Feature selection

ABSTRACT

Selection of a representative set of features is still a crucial and challenging problem in machine learning. The complexity of the problem increases when any of the following situations occur: a very large number of attributes (large dimensionality); a very small number of instances or time points (small-instance set). The first situation poses problems for machine learning algorithm as the search space for selecting a combination of relevant features becomes impossible to explore in a reasonable time and with reasonable computational resources. The second aspect poses the problem of having insufficient data to learn from (insufficient examples). In this work, we approach both these issues at the same time. The methods we proposed are heuristics inspired by nature (in particular, by biology). We propose a hybrid of two methods which has the advantage of providing a good learning from fewer examples and a fair selection of features from a really large set, all these while ensuring a high standard classification accuracy of the data. The methods used are antlion optimization (ALO), grey wolf optimization (GWO), and a combination of the two (ALO-GWO). We test their performance on datasets having almost 50,000 features and less than 200 instances. The results look promising while compared with other methods such as genetic algorithms (GA) and particle swarm optimization (PSO).

1. Introduction

One of the current challenges in feature selection is to deal with problems that involve a large number of features, sometimes in the order of millions. It is evident that a significant amount of these features are redundant and can be eliminated from the final classification process. It is also well known that fewer attributes increase the classification accuracy. From an experimental point of view, these attributes or features represent measurements of a specified entity. It is, of course, desirable to know which of these measurements have an effect on the final analysis. Reproducing the whole set of measurements can often be costly and time-consuming [1]. This aspect is usually referred to as *high-dimensionality* of the data. On the other hand, some experiments are very difficult to reproduce entirely, both due to cost and time (it may sometimes take years to get a data instance or a data time point, for example, biological experiments where a data instance could be an experiment with an animal over the duration of a number of years) [2]. This determines some datasets to have fewer instances. We refer to this aspect as a *small-instance set*.

In the field of machine learning, feature selection is one of the most studied problems. There exist a large amount of work dealing either with high-dimensional feature selection problems or with small-instance set problems. It is far more complex and challenging to deal with both these aspects at the same time and this for two reasons: the search space becomes too large to allow exploration of each state and the small number of instances do not provide sufficient examples to learn from Ref. [3].

Evolutionary computing (EC) and population-based algorithms adaptively search the feature space by using a set of search agents that interact in a social manner to reach the optimal solution [4,5]. EC methods are inspired by the animal social and biological behavior in nature like wolves, antlions, dragonflies, spiders, and so on, in a group [6].

* Corresponding author. Faculty of Computers and Information, Beni-Suef University, Egypt.
E-mail address: hossam.zawbaa@gmail.com (H.M. Zawbaa).

https://doi.org/10.1016/j.swevo.2018.02.021
Received 26 September 2016; Received in revised form 2 November 2017; Accepted 24 February 2018
Available online 1 March 2018

Cuckoo search (CS) is a heuristic search technique applied for numerous optimization problems [7]. Bat algorithm (BA) is a metaheuristic technique that uses the echolocation behavior [8]. The water cycle algorithm (WCA) is a novel metaheuristic optimization technique for constrained optimization in engineering design problems [9]. The ordinary differential equations (ODEs) is a novel algorithm applied for solving several engineering problems. The aim of this technique is to minimize the weighted residual (error) function of the ODEs [10].

Several variants use a binary version of bio-inspired algorithms [11–13] and adapt and use them in different applications [14–20].

In general, approximation methods are well suited for feature selection problems and have been applied in the past with certain success. Among those, bio-inspired heuristics show potential in proving a good classification accuracy. These methods transform the feature selection problem into a multiobjective optimization problem with two criteria: maximize the classification accuracy and minimize the number of selected features [21].

In this work, we investigate the applicability of two methods - antlion optimization (ALO) and grey wolf optimization (GWO)- for feature selection and, based on the advantages offered by each method, we propose a combination of them in a new algorithm called ALO-GWO. In the hybrid version, we exploit GWO's global search ability to find the optimal solutions (good exploration of the search space) and ALO's local search performance to get the real optimum (good exploitation of the already found solutions). It is important to ensure a good exploration/exploitation balance in the optimization process. The hybrid ALO-GWO avoids premature convergence and provides an efficient search of the feature space.

These methods are tested and evaluated on 27 different datasets. The first datasets have a balance between the number of features and the number of data instances and have the scope to test the applicability of our bio-inspired methods and the tuning of the parameters. We then perform a more challenging test on datasets having up to almost 50,000 features and fewer than 200 instances. These complex datasets are microarray experiments and image processing data. The results are compared with the ones obtained by genetic algorithms and particle swarm optimization.

Details regarding the techniques used are described in Section 3 and Section 4. Moreover, the comparison of the results is presented in Section 5. The work concluded the results and propose further directions to investigate in Section 6.

2. Related work

Various heuristic techniques mimic the behavior of biological and physical systems in nature and prove to be robust methods for global optimization. Some of them have been applied to feature selection (modeled as an optimization problem) and this section summarizes the contributions. A feature selection method based on genetic algorithm (GA) using a fuzzy set as fitness function has been proposed in [22]. With the same fitness function, particle swarm optimization (PSO) achieves better performance than GA algorithm as evident from [23]. A multi-objective method using the non-dominated sorting genetic algorithm (NSGAII) has been applied to feature selection, but the performance of the method has not been compared with any other feature selection algorithms. Hybrid NSGAII with rough sets is used as feature selection technique for microarray gene expression data [24].

A multi-objective method for feature selection based on genetic programming (GP) filter model has been applied to binary classification problems [25]. Gaussian process multi-task regression using feature selection (GPMTFS) uses the multi-tree GP technique and designs a classifier utilizing the selected features [26].

Artificial bee colony (ABC) is a numerical optimization algorithm based on the foraging behavior of honeybees [27]. A virtual bee algorithm (VBA) is applied to optimize numerical functions using a swarm of virtual bees that move randomly in the feature space and interact to find the food sources [28]. An approach based on the natural behavior of honeybees in which randomly generated worker bees are moved in the elite bee direction, the elite representing the optimal (near to optimal) solution is proposed in Ref. [29].

Ant colony optimization (ACO) based wrapper feature selection algorithm is applied to network intrusion detection [30]. ACO uses Fisher discrimination rate to adopt the heuristic information and rough set theory used for feature selection [31]. Grey wolf optimization modified to ensure a good balance between exploration and exploitation is proposed in Ref. [32]. A multi-objective variant of GWO is developed to optimize multi-objective problems [33]. New binary approaches of GWO (BGWO) are proposed and applied to feature selection problem [12]. A variant of antlion optimization using binary representation (BALO) is used for finding a feature subset that maximizes the classification performance while minimizing the number of selected features [13].

3. Preliminaries

This section provides an introduction to the standard GWO and ALO algorithms formulated for optimization.

3.1. Grey wolf optimization (GWO)

Grey wolf optimization is inspired by the hunting behavior of grey wolves in nature [34]. The grey wolves are species have a strict social dominant hierarchy of leadership, being organized into four main levels:

1. *Alpha* (α) *wolves* (one male and one female) are the leaders. They are responsible for making decisions (hunting, sleeping place, time to wake, etc.).
2. *Beta* (β) *wolves* are second-level subordinate wolves that help α in taking decisions or the other pack activities. β wolf is the best candidate to be the next α.
3. *Delta* (δ) *wolves* are the third-level subordinate and their responsibilities are scouts, sentinels, elders, hunters, and caretakers.
4. *Omega* (ω) *wolves* are the fourth-level subordinate and play the role of scapegoat.

In an optimization context, α wolves are the best solution, β and δ wolves are the second and third best solutions respectively, while ω wolves are the remaining candidate solutions. The hunting is guided by α, β, and δ, while ω follow these three candidates. In order for the pack to hunt prey, they first encircle it. The mathematical model of the encircling behavior is given in Eq. (1) [35]:

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \tag{1}$$

where $\vec{X}$ is the grey wolf position, $\vec{X}_p$ is the prey position, t is the iteration number, and $\vec{D}$ is given by:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|. \tag{2}$$

$\vec{A}$, $\vec{C}$ are coefficient vectors given by:

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a}, \tag{3}$$

$$\vec{C} = 2\vec{r_2}, \tag{4}$$

where $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in [0, 1].

The hunt is guided by α, but β and δ might also participate in hunting process. In the mathematical simulation of the hunting behavior, α, β, and δ are assumed to have better knowledge about the potential location of prey. ω update their positions according to the position of the best search agents as follows:

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}|, \overrightarrow{D_\beta} = |\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \vec{X}|, \overrightarrow{D_\delta} = |\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \vec{X}| \tag{5}$$

$$\overrightarrow{X_1} = |\overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha}|, \overrightarrow{X_2} = |\overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta}|, \overrightarrow{X_3} = |\overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta}| \quad (6)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad (7)$$

Parameter $\vec{a}$ controls the trade-off between exploitation and exploration and is linearly updated in each iteration with a range from 2 to 0 according to Eq. (8).

$$\vec{a} = 2 - t \cdot \frac{2}{\text{Max}_{Iter}}, \quad (8)$$

where Max_{Iter} is the total iteration number allowed for the optimization.

Candidate solutions tend to diverge from the prey when $|2\vec{a} \cdot \vec{r_1} - \vec{a}| > 1$ and converge towards the prey when $|2\vec{a} \cdot \vec{r_1} - \vec{a}| < 1$.

GWO is described as in Algorithm (1).

3.2. Antlion optimization

Antlion optimization has been proposed by Mirjalili [36] and mimics the hunting mechanism of antlions in nature. They often hunt in larvae and the adulthood period is for reproduction. Antlion larvae dig a hole in the sand and, after digging the trap, the larvae hide underneath the bottom of the hole and wait for insects (preferably ants) to be trapped in the hole. Once the antlion recognizes that prey is in the trap, the antlion tries to catch its prey. However, insects regularly are not caught immediately and try to escape from the trap. In this case, antlions intelligently throw sands towards to edge of the hole to slide the prey into the bottom of the hole. When prey is caught in the jaw, it is pulled from the soil and consumed. After consuming the prey, antlions throw the leftovers out of the hole and amend the hole for the next hunt.

In order to explain the steps of ALO, we use the following notations:

- t: the current iteration
- T: the total number of iterations
- $Antlion_j^t$: the position of antlion j at iteration t
- Ant_i^t: the position ant i at iteration t
- c^t: is the minimum of all variables at $t-th$ iteration.
- d^t: indicates the vector including the maximum of all variables at $t-th$ iteration.
- w: a constant defined based on the current iteration ($w = 2$ when $t > 0.1T$, $w = 3$ when $t > 0.5T$, $w = 4$ when $t > 0.75T$, $w = 5$ when $t > 0.9T$, and $w = 6$ when $t > 0.95T$). The constant w can adjust the accuracy level of exploitation.
- I: I is a ratio defined based on w using the equation

$$I = 10^w \frac{t}{T} \quad (9)$$

- a_i: minimum random walk of $i-th$ variable
- b_i: maximum random walk of $i-th$ variable.

Random walks of ants are all based on Eq. (10):

$$X(t) = \left[0, cumsum(2r(t_1) - 1); cumsum(2r(t_2) - 1); \ldots; cumsum(2r(t_T) - 1)\right], \quad (10)$$

where *cumsum* calculates the cumulative sum, T is the maximum number of iteration, t shows the step of random walk, and $r(t)$ is a stochastic function defined in Eq. (11).

Algorithm 1 Grey wolf optimization (GWO).

Input: Number of grey wolves (n), maximum iterations (Max_{Iter}).
Result: The optimal wolf position and its fitness.

1. Initialize a population of n grey wolves positions randomly.
2. Find α, β, and δ as the first three best solutions based on their fitness values.
   while $t \leq Max_{Iter}$ **do**
   foreach $Wolf_i \in pack$ **do**
   Update current wolf's position according to Eq. (1).
   end
   - Update a, A, and C as in Eqs. (2) and (3).
   - Evaluate the positions of individual wolves.
   - Update α, β, and δ positions as the first best three solutions in the current population; as in Eqs. (5), (6), and (7).
   end
3. Select the optimal grey wolf position.

The artificial antlion works in the following way: the preys are ants that move around the search space using different random walks. These random walks are influenced by the traps of antlions. Antlions build holes proportional to their fitness (the higher the fitness, the larger the hole). Antlions with larger holes have a higher probability of catching ants. Each ant can be detected by an antlion in each iteration. The range of random walks decreases adaptively to mimic sliding ants towards antlions. If an ant becomes fitter than an antlion, this indicates that it is caught and pulled under the sand by the antlion. An antlion updates its position to the latest caught prey and builds a hole to enhance its chance of catching another prey after each hunt. ALO is formulated in the Algorithm (2).

$$r(t) = \begin{cases} 1 \text{ if } rand > 0.5 \\ 0 \text{ if } rand \leq 0.5, \end{cases} \quad (11)$$

with *rand* being a random number generated with uniform distribution in the interval of $[0, 1]$.

In order to keep the random walks inside the search space, the min–max normalization is applied:

$$X_i^{t+1} = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(b_i^t - a_i)} + c_i, \quad (12)$$

Algorithm 2 Antlion optimization (ALO) algorithm.

Input: Search space, fitness function, number of ants and antlions, number of iterations (Max_{Iter}).
Result: The elitist antlion and the its fitness.

1. Initialize a population of n ant positions and n antlion positions randomly.
2. Calculate the fitness of all ants and antlions.
3. Find the fittest antlion (the elite).
4. **while** $t \leq Max_{Iter}$ **do**
   foreach ant_i **do**
      i) Select an antlion using roulette wheel (building trap); as in Eqs. (13) and (14).
      ii) Slide ants towards the antlion;as in Eqs. (15) and (16).
      iii) Create a random walk for ant_i and normalize it; as in Eq. (18).
   end
   - Calculate the fitness of all ants.
   - Replace an antlion with its corresponding ant it if becomes fitter.
   - Update elite if an antlion becomes fitter than the elite.
   end
5. Select the optimal antlion position.

Trapping of ants in antlion's hole is modeled by sliding of prey towards the selected antlion. The walk of the ant becomes bounded by the position of the antlion and is expressed in Eqs. (13), and (14):

$$c_i^t = c^t + Antlion_j^t, \tag{13}$$

$$d_i^t = d^t + Antlion_j^t. \tag{14}$$

Antlions shoot sands outwards the center of the hole once they realize that an ant is in the trap. This behavior *slides down the trapped ant* that is trying to escape. For mathematically modeling this behavior, the radius of ants' random walks hypersphere is *decreased* adaptively as shown in Eqs. (15) and (16):

$$c^t = \frac{c^t}{I}, \tag{15}$$

$$d^t = \frac{d^t}{I}. \tag{16}$$

The final stage is *catching the prey and re-building the hole* when the antlion consumes the ant. It is assumed that catching prey occur when ants become fitter (goes inside sand) than its corresponding antlion. An antlion is then expected to update its position on the latest position of the hunted ant to improve its chance of catching new prey, modeled by Eq. (17):

$$Antlion_j^t = Ant_i^t \text{ If } f(Ant_i^t) \text{ is better than } f(antlion_j^t). \tag{17}$$

To maintain the best solution(s) across iterations, elitism should be applied. The random walk of an ant is guided by the selected antlion as well as the elite antlion and therefore the repositioning of a given ant takes the form of average of both random walks as in Eq. (18).

$$Ant_i^t = \frac{R_A^t + R_E^t}{2}, \tag{18}$$

where R_A^t is the random walk around the roulette wheel selected antlion, and R_E^t is the random walk around the elite antlion.

4. The proposed hybrid ALO-GWO for feature selection

GWO is an optimization algorithm with no internal control parameter which makes it easy to use. Moreover, GWO has a rational balance between exploration (diversification) and exploitation (intensification) through the setting of the parameter a which controls the exploration/exploitation rates at each iteration. The randomly set obstacle modeler C has an impact on the walk/step scale which provides enough variation and avoids algorithm stagnation, especially in the last optimization stages. Actually, a grey wolf can be viewed as an agent that is attracted by the best three agents in the pack but with some obstacles on its way. Fig. 1 shows the walk step/scale obtained using Eq. (19) that models the attraction of an agent by one of the leader wolves:

$$\vec{X}(t+1) = \vec{X}_p(t) - (2\vec{a} \cdot \vec{r_1} - \vec{a}) \cdot |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \tag{19}$$

with the factor $(2\vec{a} \cdot \vec{r_1} - \vec{a})$ representing the walk scale/size, $\vec{C}$ representing the obstacles in the wolf's way and $\vec{X}_p(t)$ is the attractor wolf position at time t and should be replaced by the best, second best and third best solution at every optimization iteration.

As can be seen in Fig. 1, the wolf's step size has some variation while keeping a global convergence towards an *attractor* wolf -who is assumed to know more about the prey. We can also notice that the new position of the wolf (in a given dimension) may be between the attractor position and the wolf's original position which means exploration (at positive values of the step size) or may be beyond the attractor position at negative values of the step size which means exploitation. Furthermore, for step size > 1 the wolf diverges from the prey (attractor) while in the case of step size < 1 the wolf converges towards the prey. Therefore, GWO has the merit of enhanced convergence capability while keeping enough variation of the population and the optimizer can be described as goal-directed optimizer which follows the best three solutions rather than following only the best solution as in particle swarm optimization (PSO) [2].

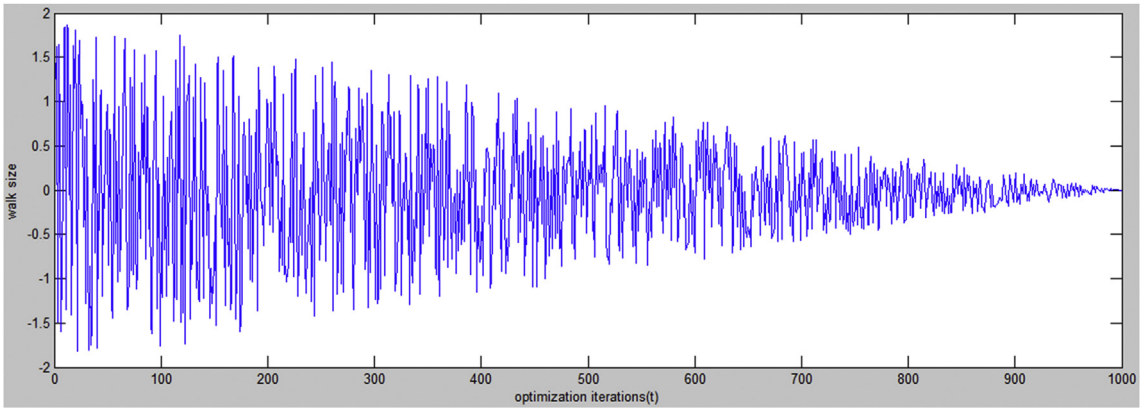

Fig. 1. GWO step size/scale for individual wolf across optimization time.

In ALO, two types of agents exist: ant and antlions. Antlions can be thought of as the set of the best solutions found ever where an antlion position is replaced only when finding better prey (ant) to replace it. Ants are in continuous motion and each ant changes its position at each iteration according to the antlions' positions. The updating mechanism of an ant relies on selecting an antlion using roulette wheel selection in combination with the best antlion and the given ant follows these two agents. Relying on an agent that is selected using roulette wheel has the merit of adding enough diversification and allows the algorithm to avoid stagnation. Stagnation avoidance capability gives such algorithm superiority over other algorithms such as PSO. Furthermore, the algorithm has minimal internal parameters compared to GA or PSO.

Choice of leader for a given swarm has a very strong impact on the explorative/exploitative capability of the algorithm. In GWO, the algorithm keeps track of the best three solutions found. So, weak agents have no chance to lead other agents and hence the exploration capability of the algorithm is lower while the exploitative capability is higher. In case of ALO, the algorithm keeps track of the best N (N is the population size) agents found and applies roulette wheel to select a leader besides the current global best to guide the swarm. So, weak agents have the chance to lead the swarm together with the current global best which enhances the exploration capability of the algorithm but at the expense of minimizing exploitation capability. An algorithm that uses both methods for leadership assignment seems to be more capable of keeping the balance between diversification and intensification. The proposed hybrid algorithm keeps both the swarm of ants in motion and also keeps the motion of antlion swarm. The hybrid algorithm updates both the weak agents (ants) using ALO principles to take the merits of ALO, as well as the strong agents (antlions) using GWO principles that have the merit of faster convergence and is more a goal follower. The proposed methodology is formally given in Algorithm (3).

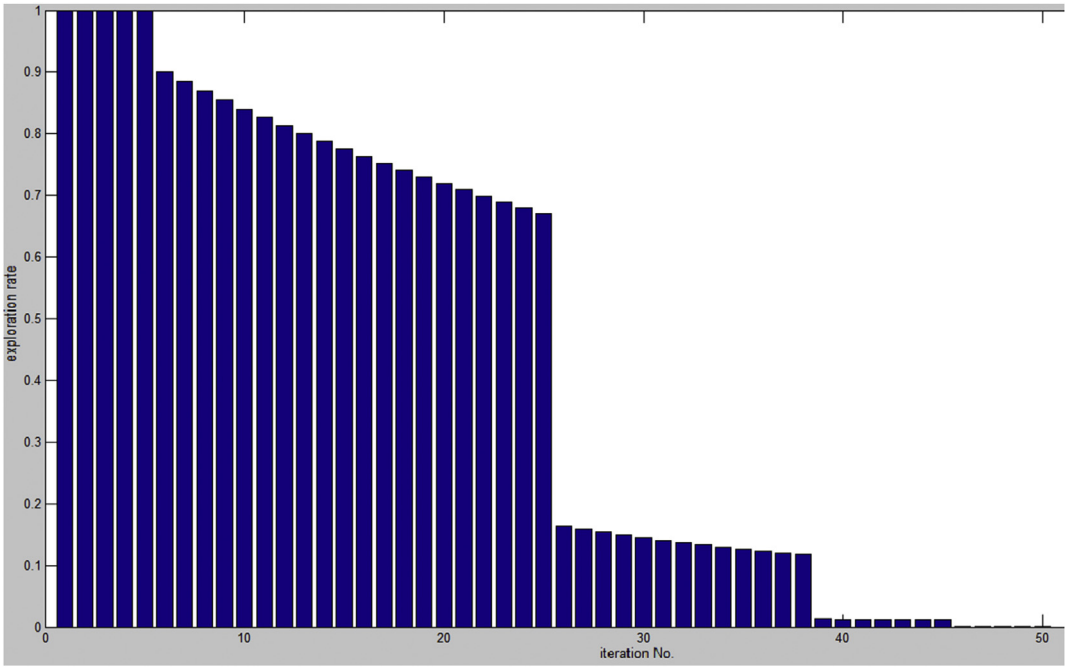

Fig. 2. Exploration rate of different ALO iterations.

Algorithm 3 The proposed hybrid ALO-GWO algorithm.

Input: Search space, fitness function, number of ants and antlions, number of iterations (Max_{Iter}).
Result: The optimal antlion position and its fitness.

1. Initialize a population of n ant positions and n antlion positions randomly.
2. Calculate the fitness of all ants and antlions.
3. Find the fittest antlion (The elite).
4. **while** $t \leq Max_{Iter}$ **do**
   foreach ant_i **do**
      i) Select an antlion using Roulette wheel (building trap).
      ii) Slide ants towards the antlion.
      iii) Create a random walk for the ant_i and normalize it.
   end
   - Calculate the fitness of all ants.
   - Replace an antlion with its corresponding ant it if becomes fitter (Catching Prey)
   - Select the *alpha*, *beta* and *delta* from the antlion population based on their fitness.
   - Update the exploration rate parameter a.
   - Update antlion positions.
   - Update the elite if an antlion becomes fitter than the elite.
   end
5. Select the optimal antlion position.

The hybrid algorithm has enough diversification capabilities as a result of:

- Roulette wheel selection of agents -drawn from ALO which impacts the ants' swarm;
- Adaptive size of the random walk as in GWO and impacts antlions swarm;
- Adaptive size of the random walks as in ALO and impacts the ants' swarm;
- The repositioning of the whole swarm rather than of ant swarm as in ALO.

Moreover, the hybrid algorithm has the following intensification capabilities:

- Goal-directed, which results from GWO walk where the three best solutions are followed and helps enhance the walk of antlions;
- Reduction of random walk scale/step size which is adopted in both GWO and ALO;
- Replacement of fitter ants with the corresponding antlion using ALO principles.

The proposed hybrid algorithm for feature selection works in a wrapper-based manner. The principal characteristic of wrapper methodologies is the use of the classification performance as a guide to feature selection procedure. K-nearest neighbor (KNN) is a supervised learning algorithm that classifies an unknown sample instance based on the majority of the K-nearest neighbor category. We use KNN as a classification method in our system to ensure the quality of the selected features [37]. The feature classification criteria or the objective function in the wrapper feature selection reflects the classification performance as well as the number of selected features as given in Eq. (20):

$$f_\theta = \alpha \cdot E + (1-\alpha)\frac{\sum_i \theta_i}{N}, \tag{20}$$

where f_θ is the fitness function of a binary vector θ of size N, with 0/1 elements representing unselected/selected features, N represents the total number of features in the dataset, E is the classifier error, and α is a constant controlling the importance of classification performance using the number of features selected.

In wrapper-based methods, the single evaluation of a given solution is very costly as it always applies training and testing of the given classier. Therefore, an effective selection of the search method is mandatory. In this study, we use a combination of ALO and GWO algorithms to adaptively search the features space, ensuring a maximization of the classification performance while keeping a minimum number of selected features. Iteratively, the antlion algorithm selects an antlion for hunting in a roulette wheel manner and performs the random walk of ants around the elite/best antlion. Based on the past two random walks, an ant adapts its location. Iteratively, if an ant becomes better than an antlion (in terms of fitness function value), the antlion eats

Table 1
List of used parameters.

Parameter	Value(s)
K for cross validation	10 or 3
M The number of runs	30
No. of search agents	8
No. of iterations	70
Problem dimension	Number of features in the data
Search domain	[0, 1]
*r*1&*r*2 in GWO & ALO	r1 and r2 drawn from the uniform distribution
Crossover Fraction in GA	0.8
Inertia factor of PSO	0.1
Individual-best acceleration factor of PSO	0.1
α parameter in the fitness function	0.99
β parameter in the fitness function	0.01

Table 2
UCI datasets.

Dataset	No. Attributes	No. Instances	No. Classes
Breastcancer	10	699	2
Exactly	13	1000	2
Exactly2	13	1000	2
Lymphography	18	148	8
M-of-n	13	1000	2
Tic-tac-toe	9	958	2
Vote	16	300	2
Zoo	17	101	7
Wine	13	178	3
Spect	22	267	2
Sonar	60	208	2
Penglung	325	73	7
Ionosphere	34	351	2
Heart	13	270	4
Congress	16	435	2
Breast	30	569	2
Chess(Krvskp)	36	3196	2
Waveform	40	5000	3

Table 3
Microarray gene expression and image datasets.

Dataset	No. Attributes	No. Instances	No. Classes
Microarray gene expression data			
CLL-SUB-111	11340	111	3
GLA-BRA-180	49151	180	4
SMK-CAN-187	19993	187	2
TOX-171	5748	171	4
GLI-85	22283	85	2
Nci9	9712	60	9
Carcinom	9182	174	11
Image (face) detection data			
ORL10P	10304	100	10
PIX10P	10000	100	10
AR10P	2400	130	10
PIE10P	2420	210	10
ORL	1024	400	40

it and changes its position to the ant's position. Antlion's update their positions according to the grey wolf principles, where the *alpha*, *beta*, and *delta* are selected among the antlions. This algorithm is iteratively applied for a number of steps. It worth mentioning that the random walk of an ant is limited by the exploration rate at the current iteration. Commonly, the exploration rate decreases as the optimization progress to allow for fine search/intensive search. The plot in Fig. 2 outlines the exploration rate of the antlion at different optimization iterations.

A similar behavior is applied to the exploration rate controlling the repositioning of antlions, where antlions' exploration rate is linearly decreased from 2 to 0 at each iteration in order to allow for fine search.

An individual solution is represented as a continuous-valued vector with the same dimension as the number of attributes in the given dataset. The solution vector is limited to the range [0, 1]. For a solution evaluation, the continuous values are mapped to a binary representation using Eq. (21):

$$y_{ij} = \begin{cases} 0 \text{ If}(x_{ij} < 0.5) \\ 1 \text{ Otherwise,} \end{cases} \tag{21}$$

where x_{ij} is the continuous value of solution i in dimension j, and y_{ij} is the discrete representation of solution vector x.

ALO-GWO works on the wrapper-based feature selection manner employing KNN in classification data and ELM in regression data. The

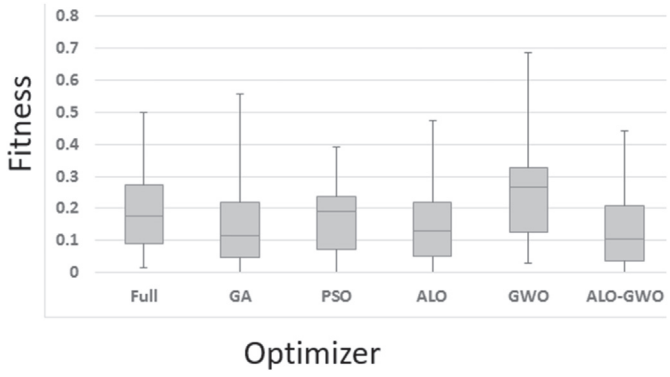

Fig. 3. Mean, best, worst, and standard deviation fitness function values obtained by all optimizes using small initialization.

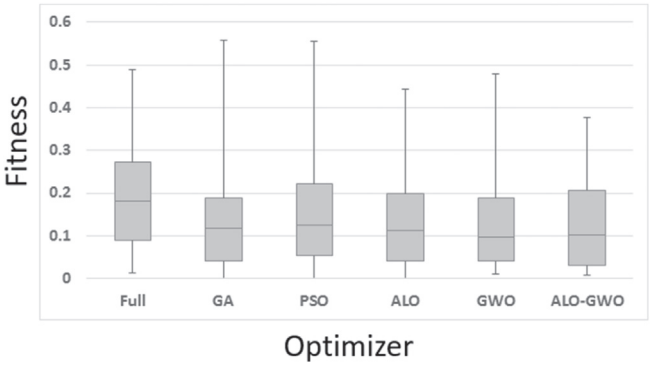

Fig. 4. Mean, best, worst, and standard deviation fitness function values obtained by all optimizes using uniform initialization.

running time may increase when changing to another classifier, for example, support vector machine (SVM), random forest (RF), or artificial neural network (ANN). Therefore, switching to different classification methods should be carefully handled, particularly if the algorithm is adopted in real-time applications. The main limitation of the methodology presented in this paper is the non-exact repeatability of the hybrid ALO-GWO results. We monitored this at various applications of the algorithm. The subset of features selected might not be the same. Despite the fact that the resulting solutions are all good solutions, it might be confusing for the user to figure out which subset to consider.

5. Experimental results, analysis and discussions

5.1. Datasets used for experiments

The global and optimizer-specific parameter settings are outlined in Table 1. All the parameters are set either according to domain-specific knowledge as in the case of α, β parameters, based on trial and error on small simulations, or common in the literature such as the rest of the parameters.

We first tested the performance of ALO, GWO, and the hybrid ALO-GWO on eighteen datasets taken from the UCI machine learning repository [38]. These datasets together with their number of features and the number of instances are shown in Table 2. We can observe that these datasets have a low ratio between the number of features and the number of instances. We then moved to more complex experiments for which the ratio between the number of attributes and the number of features is really high. These datasets (given in

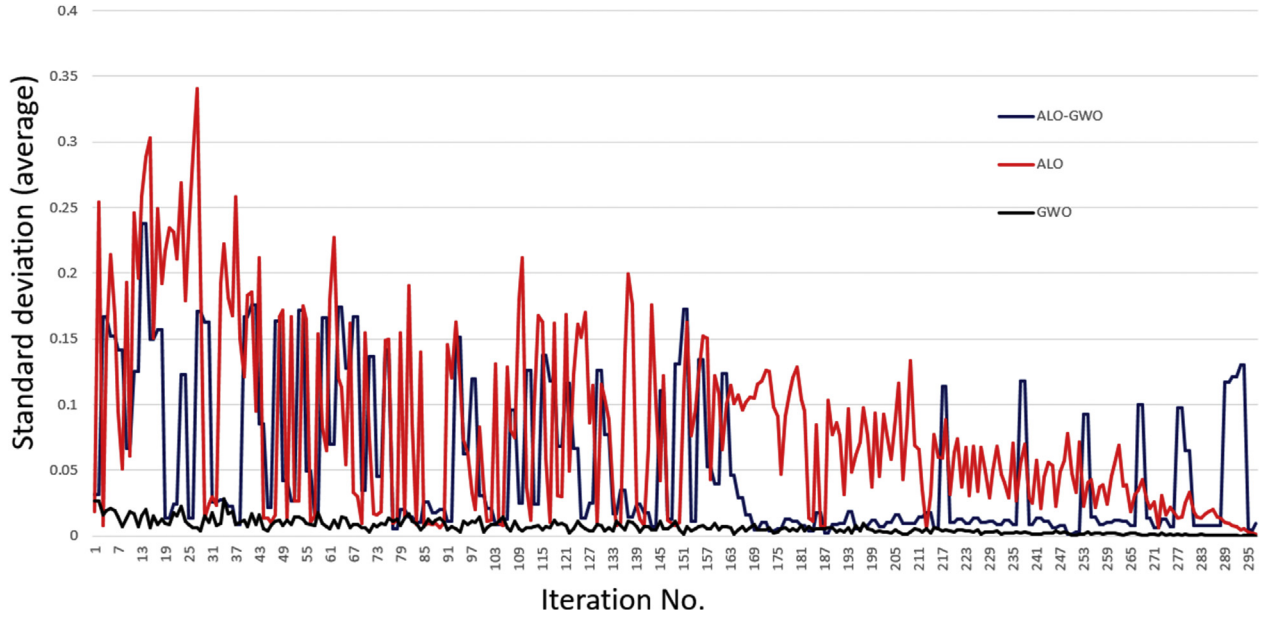

Fig. 5. Population diversity for GWO, ALO, and ALO-GWO.

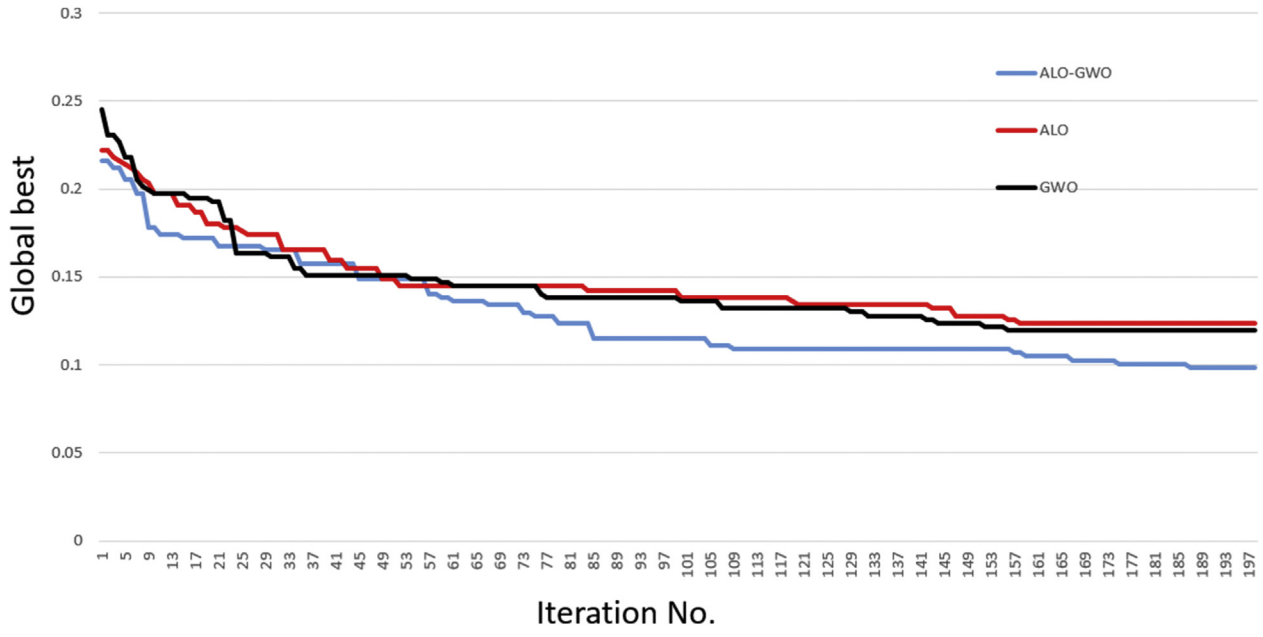

Fig. 6. Global best fitness (average) for GWO, ALO, and ALO-GWO.

Table 3) are taken from [39]. Seven of these datasets are microarray gene expression data and the other five are image (face) detection data.

10-fold cross-validation is used for the data in Table 2 where one fold is kept for *testing* and the remaining 9-folds are equally divided between *training* and *validation* portions. 3-fold cross-validation is used for the data in Table 3 where an individual dataset is divided randomly into three equal portions namely *training*, *validation*, and *testing* [40]. The validation set is used as the test set at the feature selection optimization time. The test set is hidden until the final evaluation of the selected features, the training data is used for training the classifier at both feature selection optimization and at the final testing stage. Such procedure is repeated *M* times to accurately assess statistical evaluation indicators.

5.2. Evaluation criteria

The well-known KNN is used as a classifier to evaluate the performance of individual algorithms with k = 5 [37]. Each optimization algorithm is repeated *M* times with different random seeds to test convergence capability.

We use a set of quantitative measures in order to analyze the results obtained by the methods we apply. The first three metrics are used to measure the mean, best, and worst expected performance of the algo-

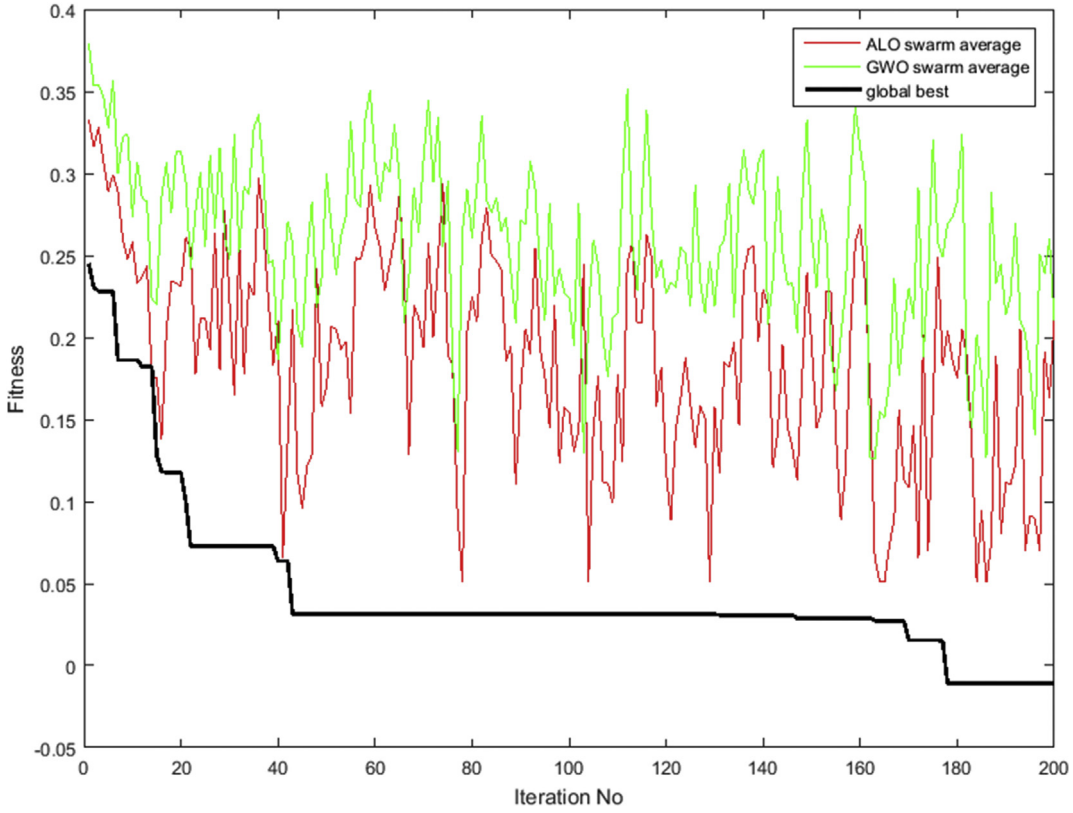

Fig. 7. Swarm mean fitness after applying ALO and GWO with the global best at each iteration.

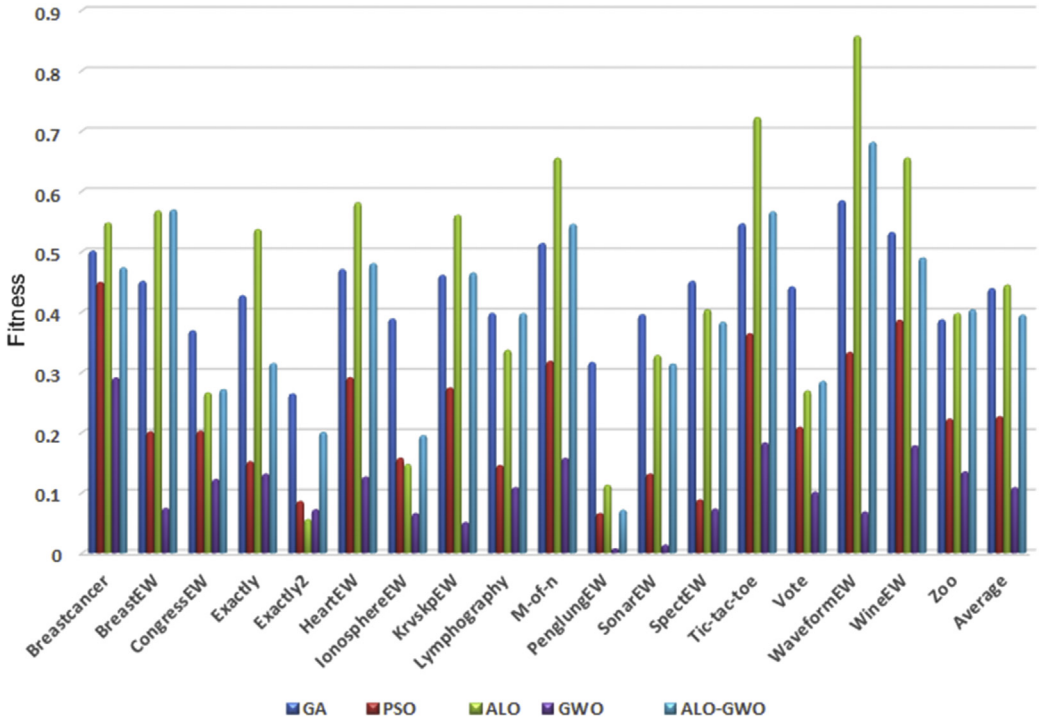

Fig. 8. Average selected feature values using small initialization.

rithms. The fourth measure shows the ability of the optimizer to converge to the same optimal solution. The fifth metric shows the performance of the classifier on test data. The sixth and seventh metrics are a measure of the size of selected features set (we expect a low number of features and low classification error) and the resulted data compactness/separability after applying feature selection. A final indicator is used to assess the statistical significance of the difference between two optimizers.

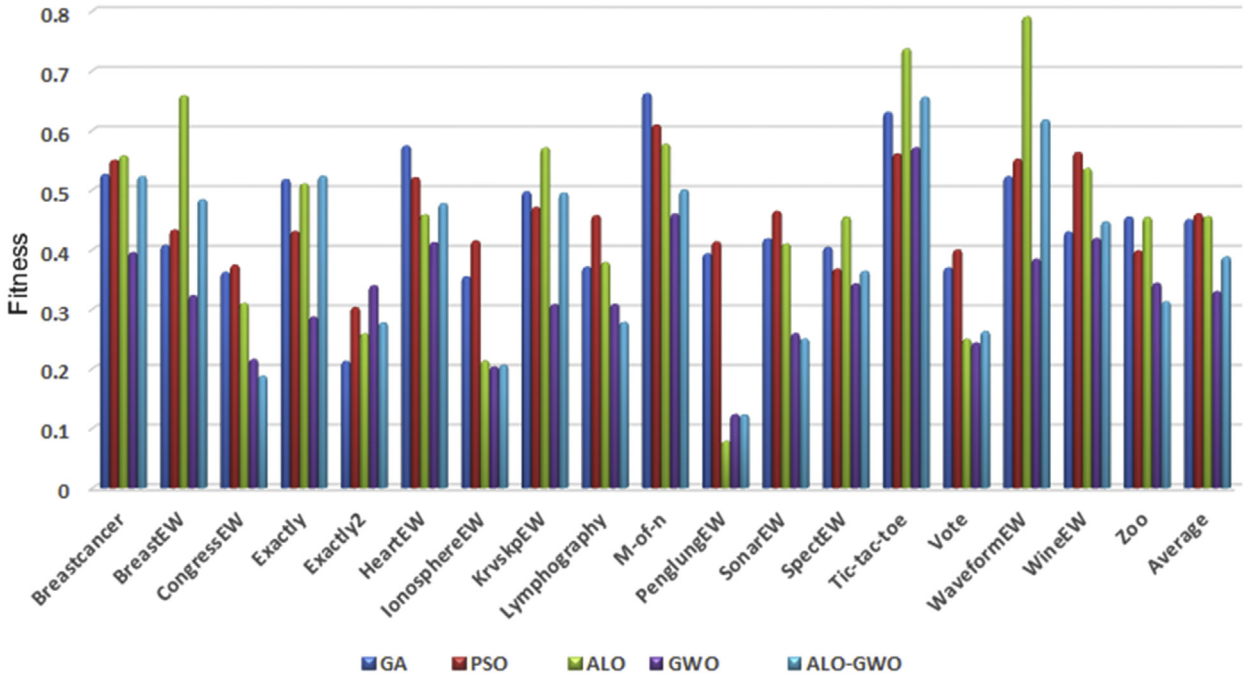

Fig. 9. Average selected feature values using uniform initialization.

1. **Mean fitness:** is an average value of all the solutions in the final sets obtained by an optimizer in a number of individual runs [41].
2. **Best fitness:** is the best solution found by an optimizer in all the final sets resulted from a number of individual runs [41].
3. **Worst fitness:** is the worst solution found by an optimizer in all the final sets resulted from a number of individual runs [41].
4. **Standard deviation (Std):** is used to ensure that the optimizer convergence to the same optimal and ensures repeatability of the results. It is computed over all the sets of final solutions obtained by an optimizer in a number of individual runs [42].
5. **Classification mean square error (CMSE):** is a measure of classifier's average performance on the test data. It is averaged over all final sets in all the independent runs [43].
6. **Average selected features:** represents the average ratio of the selected features subset to the original number of features. The average is computed for each final set of solutions in multiple individual runs.
7. **Average Fisher score:** evaluates a feature subset such that in the data space spanned by the selected features, the distances between data points in different classes are as large as possible, while the distances between data points in the same class are as small as possible [44]. Fisher score in this work is calculated for individual features given the class labels; as follows:

$$F_j = \frac{\sum_{k=1}^{c} n_k(\mu_k^j - \mu^j)^2}{(\sigma^j)^2}, \tag{22}$$

where F_j is the Fisher index for feature j, μ^j and $(\sigma^j)^2$ are the mean and std of the dataset, n_k is the size of class k, and μ_k^j is the average of class k. The Fisher for a set of features is defined as:

$$F_{tot} = \frac{1}{S}\sum_{i=1}^{S} F_i, \tag{23}$$

Table 4
Average Fisher score values using small initialization.

Dataset	GA	PSO	ALO	GWO	ALO-GWO
Breastcancer	0.654	0.638	**0.730**	0.408	0.648
Breast	0.235	0.128	0.285	0.053	**0.306**
Congress	**0.208**	0.129	0.169	0.073	0.171
Exactly	**0.001**	0	**0.001**	0	**0.001**
Exactly2	**0.001**	0	0	0	**0.001**
Heart	0.079	0.053	**0.093**	0.025	0.079
Ionosphere	**0.033**	0.015	0.013	0.005	0.021
Chess(Krvskp)	0.019	0.011	**0.021**	0.003	**0.021**
Lymphography	**0.157**	0.062	0.137	0.020	0.130
M-of-n	0.027	0.020	**0.031**	0.009	**0.031**
Penglung	**0.253**	0.061	0.088	0.007	0.069
Sonar	**0.019**	0.008	0.016	0.002	0.015
Spect	**0.027**	0.005	0.023	0.002	0.023
Tic-tac-toe	0.005	0.004	**0.006**	0.001	0.005
Vote	0.190	0.112	0.161	0.072	**0.163**
Waveform	0.124	0.086	**0.152**	0.019	0.142
Wine	0.538	0.485	**0.701**	0.250	0.558
Zoo	11.136	6.925	11.195	3.744	**12.899**

Table 5
Average Fisher score values using uniform initialization.

Dataset	GA	PSO	ALO	GWO	ALO-GWO
Breastcancer	0.717	0.750	0.717	0.559	**0.765**
Breast	0.220	0.230	**0.313**	0.189	0.268
Congress	0.195	**0.203**	0.183	0.157	0.163
Exactly	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Exactly2	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Heart	**0.094**	0.084	0.078	0.072	0.077
Ionosphere	0.028	**0.038**	0.019	0.015	0.028
Chess(Krvskp)	0.021	0.020	**0.022**	0.019	0.020
Lymphography	0.118	0.178	**0.146**	0.060	0.119
M-of-n	0.030	0.029	**0.031**	0.029	**0.031**
Penglung	**0.304**	0.303	0.062	0.099	0.097
Sonar	0.019	**0.020**	0.018	0.014	0.013
Spect	0.024	0.022	**0.025**	0.022	0.021
Tic-tac-toe	**0.006**	0.005	**0.006**	0.005	0.005
Vote	0.192	**0.199**	0.144	0.150	0.156
Waveform	0.129	0.128	**0.154**	0.114	0.139
Wine	0.482	0.532	**0.548**	0.500	0.503
Zoo	12.573	11.229	**13.617**	11.650	10.635

Table 6
Wilcoxon test on the average fitness using small initialization.

Optimizers	Wilcoxon Test value	Comment
ALO-GWO - ALO	0.002	significant
ALO-GWO - GWO	0.04	significant
ALO-GWO - PSO	0	significant
ALO-GWO - GA	0.003	significant

Table 7
Wilcoxon test on the average fitness using uniform initialization.

Optimizers	Wilcoxon Test value	Comment
ALO-GWO - ALO	0.006	significant
ALO-GWO - GWO	0.06	insignificant
ALO-GWO - PSO	0	significant
ALO-GWO - GA	0.002	significant

where S is the number of selected features. The average Fisher score over a set of N runs is defined as:

$$Fishr - score = \frac{1}{N}\sum_{i=1}^{N} F_{tot}^{i}, \tag{24}$$

where F_{tot}^{i} is the Fisher score computed for selected feature set on run i.

8. **Wilcoxon rank sum test:** is a nonparametric test for significance assessment. The test assigns rank to all the scores considered as one group and then sums the ranks of each group [45]. The test statistic relays on calculating W as in Eq. (25):

$$W = \sum_{i=1}^{N} (sgn(x_{2,i} - x_{1,i}).R_i), \tag{25}$$

where $x_{2,i}$, $x_{1,i}$ are the best fitness values obtained by first and second optimizer on run i, R_i is the rank of difference between $x_{2,i}$ and $x_{1,i}$, and $sgn(x)$ is the standard sign function.

5.3. Results and discussion on UCI datasets

Two major scenarios are used to benchmark the proposed hybrid model versus the other methods depending on the initialization of search agents', namely uniform and small initialization. In uniform initialization, the search agents are positioned randomly on the search space following the uniform distribution random number generator (RNG). In small initialization, the search agents are initialized with a minor number of features selected. The purpose of using this initialization is to test whether certain algorithms can reach the optimum even if the initial population is very different from the optimum.

Table 8
Best fitness obtained using uniform initialization for the microarray and image detection datasets.

Dataset	ALO-GWO	ALO	GWO	GA	PSO	Full
Microarray gene expression data						
CLL-SUB-111	**0.163**	0.189	0.165	0.188	**0.163**	0.242
GLA-BRA-180	0.151	**0.150**	0.185	0.283	0.266	0.383
SMK-CAN-187	**0.118**	0.145	0.128	0.175	0.206	0.402
TOX-171	**0.192**	0.247	0.158	0.245	0.311	0.546
GLI-85	**0**	**0**	**0**	**0**	**0**	0.103
Nci9	**0**	0.1	0.3	0.389	0.389	0.444
Carcinom	**0.017**	**0.017**	0.034	0.052	0.034	0.086
Image (face) detection data						
ORL10P	**0**	**0**	**0**	0.059	0.058	0.152
PIX10P	0.115	**0.097**	0.167	0.232	0.239	0.229
AR10P	**0.155**	0.171	0.156	0.230	0.228	0.512
PIE10P	0.161	0.173	**0.157**	0.231	0.227	0.229
ORL	**0.178**	0.204	**0.178**	0.216	0.231	0.313

Table 9
Worst fitness obtained using uniform initialization for the microarray and image detection datasets.

Dataset	ALO-GWO	ALO	GWO	GA	PSO	Full
Microarray gene expression data						
CLL-SUB-111	**0.352**	0.429	0.356	0.461	0.430	0.651
GLA-BRA-180	0.372	**0.278**	0.387	0.538	0.559	0.604
SMK-CAN-187	**0.300**	0.311	0.415	0.414	0.523	0.618
TOX-171	**0.383**	0.417	0.420	0.475	0.592	0.789
GLI-85	**0.31**	**0.31**	0.414	0.483	0.483	0.483
Nci9	**0.7**	0.8	0.824	0.85	0.824	0.85
Carcinom	**0.31**	**0.31**	0.345	**0.31**	0.345	0.379
Image (face) detection data						
ORL10P	**0.304**	0.381	0.379	0.471	0.467	0.486
PIX10P	0.369	**0.358**	0.414	0.526	0.526	0.397
AR10P	**0.602**	0.736	0.614	0.778	0.775	0.805
PIE10P	**0.453**	0.488	0.485	0.507	0.551	0.538
ORL	0.481	**0.444**	0.459	0.451	0.466	0.519

Table 10
Mean fitness obtained using uniform initialization for the microarray and image detection datasets.

Dataset	ALO-GWO	ALO	GWO	GA	PSO	Full
Microarray gene expression data						
CLL-SUB-111	**0.252**	0.285	0.277	0.323	0.341	0.424
GLA-BRA-180	**0.230**	**0.230**	0.297	0.353	0.378	0.499
SMK-CAN-187	**0.214**	0.217	0.221	0.270	0.304	0.507
TOX-171	0.287	0.324	**0.286**	0.365	0.440	0.654
GLI-85	**0.077**	0.086	0.138	0.144	0.143	0.233
Nci9	0.491	**0.481**	0.597	0.616	0.613	0.679
Carcinom	**0.144**	0.149	0.168	0.170	0.168	0.248
Image (face) detection data						
ORL10P	**0.180**	0.200	0.248	0.305	0.305	0.330
PIX10P	**0.239**	0.349	0.386	0.477	0.494	0.259
AR10P	0.514	**0.494**	0.503	0.604	0.626	0.702
PIE10P	**0.295**	0.329	0.318	0.392	0.399	0.401
ORL	**0.327**	0.333	0.319	0.340	0.339	0.419

Mean, best, worst, and standard deviation fitness function values obtained by all the optimizers using small initialization are presented in Fig. 3, while the ones using the uniform initialization are presented in Fig. 4. We can observe that the hybrid ALO-GWO outperforms GA, PSO, GWO, and ALO, which proves the capability of ALO-GWO to adaptively search the feature space for optimal feature combination and avoid premature convergence that may be caused by stagnation in local minima. The enhanced performance of the hybrid algorithm can be justified by the fact that the algorithm takes advantages of GWO such as being more exploitative by following the three best solutions, and also the advantage of ALO for adopting roulette wheel selection which provides more diverse search capability and hence helps to avoid premature convergence. Moreover, we can see that the hybrid algorithm uses random step sizes for the random walk but within a linearly decremented envelope. The random setting of the random walk scale helps the optimizer to avoid stagnation and provides diverse solutions. Fig. 1 depicts the adopted random walk scale used in the hybrid algorithm. We can remark also that the enhanced performance of the proposed hybrid algorithm is comparable for both small and uniform initialization which proves the capability to generate enough diversity in the population, which is a result of repositioning both the ants and the antlions rather than the repositioning the half of the swarm. Moreover, the diversity of population results from the adoption of the roulette

Table 11
Standard deviation fitness obtained using uniform initialization or the microarray and image detection datasets.

Dataset	ALO-GWO	ALO	GWO	GA	PSO	Full
Microarray gene expression Data						
CLL-SUB-111	**0.053**	0.058	0.059	0.073	0.069	0.094
GLA-BRA-180	0.055	**0.043**	0.049	0.056	0.067	0.057
SMK-CAN-187	**0.037**	0.042	0.062	0.055	0.074	0.054
TOX-171	**0.046**	0.053	0.072	0.061	0.068	0.071
GLI-85	**0.057**	0.061	0.083	0.086	0.086	0.089
Nci9	0.118	0.120	0.106	0.105	0.105	**0.103**
Carcinom	0.067	**0.065**	0.067	**0.065**	0.067	0.071
Image (face) detection data						
ORL10P	**0.064**	0.096	0.093	0.104	0.100	0.083
PIX10P	**0.075**	0.086	0.106	0.117	0.116	0.097
AR10P	**0.071**	0.085	0.077	0.086	0.080	0.081
PIE10P	**0.078**	0.090	0.083	0.085	0.091	0.089
ORL	**0.045**	0.048	0.053	0.049	0.046	0.050

Table 13
Classification mean square error obtained using uniform initialization.

Dataset	ALO-GWO	ALO	GWO	GA	PSO	Full
Microarray gene expression data						
CLL-SUB-111	**0.257**	0.281	0.271	0.321	0.336	0.416
GLA-BRA-180	**0.224**	**0.224**	0.294	0.355	0.374	0.495
SMK-CAN-187	0.232	**0.216**	0.229	0.269	0.308	0.505
TOX-171	0.287	0.325	**0.281**	0.359	0.443	0.655
GLI-85	**0.221**	0.223	0.226	0.226	0.230	**0.221**
Nci9	0.683	0.707	0.686	0.675	0.675	**0.663**
Carcinom	**0.231**	0.238	0.234	0.236	0.232	0.236
Image (face) detection data						
ORL10P	**0.183**	0.202	0.246	0.309	0.305	0.321
PIX10P	0.117	**0.097**	0.161	0.231	0.236	0.256
AR10P	**0.474**	0.489	0.509	0.599	0.615	0.695
PIE10P	**0.293**	0.328	0.317	0.402	0.404	0.402
ORL	0.406	0.404	**0.402**	0.408	0.409	0.429

wheel in the selection process of antlions. Fig. 5 depicts a simple diversity measure for all search agents for GWO, ALO, and the proposed hybrid ALO-GWO. The diversity measure calculates the standard deviation of the agents' positions averaged over all dimensions. We can see from the figure the high diversification of ALO-GWO which results from using roulette wheel selection, the random setting of random walk scale and the change of positions of all ants and antlions in every iteration.

Fig. 7 depicts the swarm mean fitness at each iteration after applying the ALO step and the GWO step as well as the global best fitness on a sample test data. We can remark from the figure that ALO occasionally squashes some search agent apart from the current swarm local region (explorative agents) and hence we can see that it occasionally worsens the swarm mean fitness. On the contrary, we see that GWO with its exploitative behavior always attracts the swarm towards the best-performing agents (alpha, beta, and delta) and hence enhances the swarm mean fitness. By combing the two we can see that the global best fitness can avoid stagnation and keeps a good performance.

For assessing the stability of the stochastic algorithms and the capability of converging to the same/similar optimum, we measure the standard deviation of the obtained fitness values over the different runs. We can again observe that the minimum for this measure is obtained by ALO-GWO in the uniform initialization, while for the small initialization ALO-GWO and ALO have comparable standard deviation. Having a minimum value for the standard deviation measure proves the capability of the optimizer to abandon local optima and to converge to the global optimum. Again, such capability can be interpreted by the enhanced diversification of the optimizer which enables it to avoid premature convergence. Fig. 6 highlights the convergence curve (on average) for one of the datasets using the different optimizers and one can notice that the hybrid algorithm still has the capability to find a better solution at the final optimization stages while both ALO and GWO stagnate at local minima.

Table 14
Wilcoxon test on the average fitness for different optimizers using uniform initialization.

Optimizers	Wilcoxon Test value	Comment
ALO-GWO - ALO	0.630	insignificant
ALO-GWO - GWO	0.001	significant
ALO-GWO - PSO	0	significant
ALO-GWO - GA	0	significant

Table 12
Average size of the selected feature set using uniform initialization for the microarray and image detection datasets.

Dataset	ALO-GWO	ALO	GWO	GA	PSO
Microarray gene expression data					
CLL-SUB-111	**0.110**	0.132	0.110	0.488	0.482
GLA-BRA-180	0.082	**0.018**	0.142	0.495	0.491
SMK-CAN-187	**0.091**	0.092	0.163	0.491	0.488
TOX-171	0.174	0.204	**0.163**	0.488	0.481
GLI-85	0.211	**0.206**	0.403	0.500	0.500
Nci9	0.137	**0.119**	0.370	0.500	0.500
Carcinom	0.295	**0.278**	0.412	0.499	0.499
Image (face) detection data					
ORL10P	0.030	**0.005**	0.082	0.480	0.480
PIX10P	**0.002**	0.003	0.072	0.479	0.479
AR10P	**0.025**	0.057	0.115	0.472	0.462
PIE10P	**0.065**	0.068	0.115	0.474	0.462
ORL	0.340	**0.331**	0.346	0.499	0.502

Regarding the size of selected features with respect to the original size, Figs. 8 and 9 present this ratio. We can observe that, although ALO-GWO outperforms all other methods in classification performance, has a comparable ratio of features selected which confirms that ALO-GWO can select the optimal feature combination with comparable size. Fisher score is calculated for the output of the different feature selection systems over the different datasets as shown in Tables 4 and 5. We can again observe that the best Fisher score value is achieved by ALO-GWO using small initialization. This confirms that the performance of ALO-GWO is better compared to the other methods used.

Furthermore, we measured the Wilcoxon test for all the algorithms and the results are presented in Tables 6 and 7. We can see that the score obtained by ALO-GWO is significantly better compared to ALO (0.002), GWO (0), PSO (0), and GA (0.002) using small initialization. While using uniform initialization, ALO-GWO obtains significantly better results compared to ALO (0.005), PSO (0), GA (0.002), and GWO (0.0.06).

5.4. Results and discussion on microarray and image datasets

Table 8 shows the best fitness function values obtained by all optimizers for the five microarray datasets and the four image datasets. We observe that ALO-GWO outperforms GA, PSO, GWO, and ALO for the datasets that have lower attributes, such as ORL10P, AR10P, and PIE10P. GWO outperforms GA, PSO, ALO, and ALO-GWO for the datasets that have a larger number of attributes. Table 9 outlines the worst fitness function values obtained by all the algorithms for all the nine datasets. Table 10 presents the mean fitness function values. We remark that the results of ALO-GWO on average, outperform the results

obtained by GA, PSO, ALO, and GWO. Such a huge search space with a large number of features (dimensions) adds more difficulty on the optimizer. Again, the advantages of using the hybrid algorithm become more apparent and the added diversification, as well as intensification capabilities, are clearer.

Moreover, we measured the standard deviation of the solutions obtained by all the algorithms, and the results are reported in Table 11. The minimum value of the *std* measure is obtained by ALO-GWO. Regarding the size of selected features with respect to the original feature set size, we can see from Table 12 that, although ALO-GWO outperforms all the other methods in reduction rate, has a comparable ratio of features selected, which confirms that the ALO-GWO can select the optimal feature combination with comparable size. From Table 13, we can observe that ALO-GWO achieves the best classification performance.

Furthermore, we measured the Wilcoxon test for all the optimizers and the results are reported in Table 14. The results obtained by ALO-GWO are significantly better while compared with GWO (0.001), PSO (0), and GA (0), and insignificant while compared to GWO (0.630).

6. Conclusion and future work

The aim of this work is to investigate and analyze the potential of antlion optimization, grey wolf optimization, and hybrid antlion-grey wolf algorithm for selecting significant features from datasets in which the number of attributes is very large while the number of instances is relatively small. This type of feature selection datasets are challenging for machine learning algorithms as the complexity of the search space is very large due to a large number of features, on one hand, while on the other hand, the small number of instances do not provide sufficient information for learning.

We concentrate our efforts on proving that the nature-inspired heuristics can be adapted to perform very well in these situations. The two algorithms investigated here are hybridized in a third model and the performance of these models seems to outperform that of the other models used for comparisons such as genetic algorithms and particle swarm optimization. This indicates that these methods are efficient optimizers for large-dimensional small-instance datasets and are able to obtain accurate results in terms of classification on feature subset selection. The hybrid ALO-GWO algorithm has a very good balance between the exploration of the large search space and the exploitation of optimal solutions. The algorithm has the capability to generate diverse solutions and to avoid premature convergence.

On the basis of future performance, we have a few ideas that can be investigated in addition to the work presented here:

1. Use enhanced initialization method that starts the optimization with solution pool closer to optimal;
2. Extend the hybrid algorithm to work on parallel distributed mode to enhance convergence time;
3. Test the methodology on other big datasets besides those from bioinformatics.

Acknowledgment

This work was supported by the IPROCOM Marie Curie initial training network, funded through the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/under REA grant agreement No. 316555.

References

[1] I.S. Oh, J.S. Lee, B.R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1424–1437.
[2] X. Bing, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671.
[3] I. Guyon, A. Elisseeff, An introduction to variable and attribute selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[4] X.S. Yang, Nature-inspired Metaheuristic Algorithms, second ed., Luniver Press, UK, 2010.
[5] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[6] F. Valdez, Bio-inspired Optimization Methods, Springer Handbook of Computational Intelligence, 2015, pp. 1533–1538.
[7] X.S. Yang, S. Deb, Cuckoo search via levy flights, in: World Congress on Nature and Biologically Inspired Computing, 2009.
[8] X.S. Yang, A New Metaheuristic Bat-inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence, vol. 284, Springer, 2010, pp. 65–74.
[9] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[10] A. Sadollah, Y. Choi, D.G. Yoo, J.H. Kim, Metaheuristic algorithms for approximate solution to ordinary differential equations of longitudinal fins having various profiles, Appl. Soft Comput. 33 (2015) 360–379.
[11] M.A.K. Azad, A.M.A.C. Rocha, E.M.G.P. Fernandes, Improved binary artificial fish swarm algorithm for the 0-1 multidimensional knapsack problems, Swarm Evol. Comput. 14 (2014) 66–75.
[12] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[13] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing, Elsevier 213 (November 2016) 54–65.
[14] X.B. Meng, X.Z. Gao, L. Lu, Y. Liu, H. Zhang, A New Bio-inspired Optimisation Algorithm: Bird Swarm Algorithm, Experimental and Theoretical Artificial Intelligence, 2015, pp. 1–15.
[15] H. Chiroma, N.L.M. Shuib, S.A. Muaz, A.I. Abubakar, L.B. Ila, J.Z. Maitama, A review of the applications of bio-inspired flower pollination algorithm, in: International Conference on Soft Computing And Software Engineering, Procedia Computer Science, vol. 62, 2015, pp. 435–441.
[16] E. Emary, H.M. Zawbaa, C. Grosan, Experienced grey wolf optimizer through reinforcement learning and neural networks, IEEE Trans. Neural Netw. Learn. Syst. (TNNLS) 29 (3) (March 2018) 681–694.
[17] E. Emary, H.M. Zawbaa, Impact of chaos functions on modern swarm optimizers, PLoS One 11 (7) (July 2016) e0158738.
[18] H.M. Zawbaa, J. Szlk, C. Grosan, R. Jachowicz, A. Mendyk, Computational intelligence modeling of the macromolecules release from PLGA microspheres-focus on feature selection, PLoS One 11 (6) (June 2016) e0157610.
[19] E. Emary, H.M. Zawbaa, A.E. Hassanien, B. Parv, Multi-objective retinal vessel localization using flower pollination search algorithm with pattern search, Adv. Data Anal. Classif. Springer 11 (3) (September 2017) 611–627.
[20] H.M. Zawbaa, E. Emary, C. Grosan, Feature selection via chaotic antlion optimization, PLoS One 11 (3) (March 2016) e0150652.
[21] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for attribute selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[22] B. Chakraborty, Genetic algorithm with fuzzy fitness function for feature selection, in: International Symposium on Industrial Electronics, vol. 1, IEEE, 2002, pp. 315–319.
[23] A.E. Eiben, P.E. Raue, Z. Ruttkay, Genetic algorithms with multi-parent recombination, in: PPSN III: Proceedings of the International Conference on Evolutionary Computation, the Third Conference on Parallel Problem Solving from Nature, 1994, pp. 78–87.
[24] T.M. Hamdani, J.M. Won, A.M. Alimi, F. Karray, Multi-objective feature selection with NSGA II, in: Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science, vol. 4431, Springer, 2007, pp. 240–247.
[25] K. Neshatian, M. Zhang, Genetic programming for feature subset ranking in binary classification problems, in: Genetic Programming, Springer, 2009, pp. 121–132.
[26] D. Muni, N. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, IEEE Trans. Syst. Man Cybern. B Cybern. 36 (1) (2006) 106–117.
[27] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[28] X.S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, in: Lecture Notes in Computer Science, Springer (GmbH), 2005, pp. 317–323.
[29] K. Sundareswaran, V.T. Sreedevi, Development of novel optimization procedure based on honey bee foraging behavior, in: IEEE International Conference on Systems, Man and Cybernetics, 2008, pp. 1220–1225.
[30] H.H. Gao, H.H. Yang, X.Y. Wang, Ant colony optimization based network intrusion feature selection and detection, in: International Conference on Machine Learning and Cybernetics, vol. 6, 2005, pp. 3871–3875.
[31] H. Ming, A rough set based hybrid method to feature selection, in: International Symposium on Knowledge Acquisition and Modeling, 2008, pp. 585–588.
[32] N. Mittal, U. Singh, B.S. Sohi, Modified grey wolf optimizer for global engineering optimization, in: Applied Computational Intelligence and Soft Computing, vol. 2016, 2016.
[33] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.D.S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119.

[34] Emary, E., Zawbaa, H.M., Grosan, C., Hassanien, A.E., Feature subset selection approach by Gray-wolf optimization. 1st International Afro-European Conference For Industrial Advancement (AECIA), Addis Ababa, Ethiopia, 17-19 November, 2014.
[35] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software, vol. 69, 2014, pp. 46–61.
[36] S. Mirjalili, The Ant Lion Optimizer, Advances in Engineering Software, 2015, pp. 83–98.
[37] L.Y. Chuang, H.W. Chang, C.J. Tu, C.H. Yang, Improved binary PSO for feature selection using gene expression data, Comput. Biol. Chem. 32 (2008) 29–38.
[38] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010.
[39] Feature selection, http://featureselection.asu.edu/index.php, last visit on 25 November 2015.
[40] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[41] S.L. Tilahun, H.C. Ong, Prey-predator algorithm: a new metaheuristic algorithm for optimization problems, Inf. Technol. Decis. Making 14 (6) (2015) 1331–1352.
[42] L. Jia, W. Gong, H. Wu, An improved self-adaptive control parameter of differential evolution for global optimization, Comput. Intell. Intell. Syst. 51 (2009) 215–224.
[43] E.L. Lehmann, G. Casella, Theory of Point Estimation, second ed., Springer, New York, 1998.
[44] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience Publication, 2000.
[45] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.