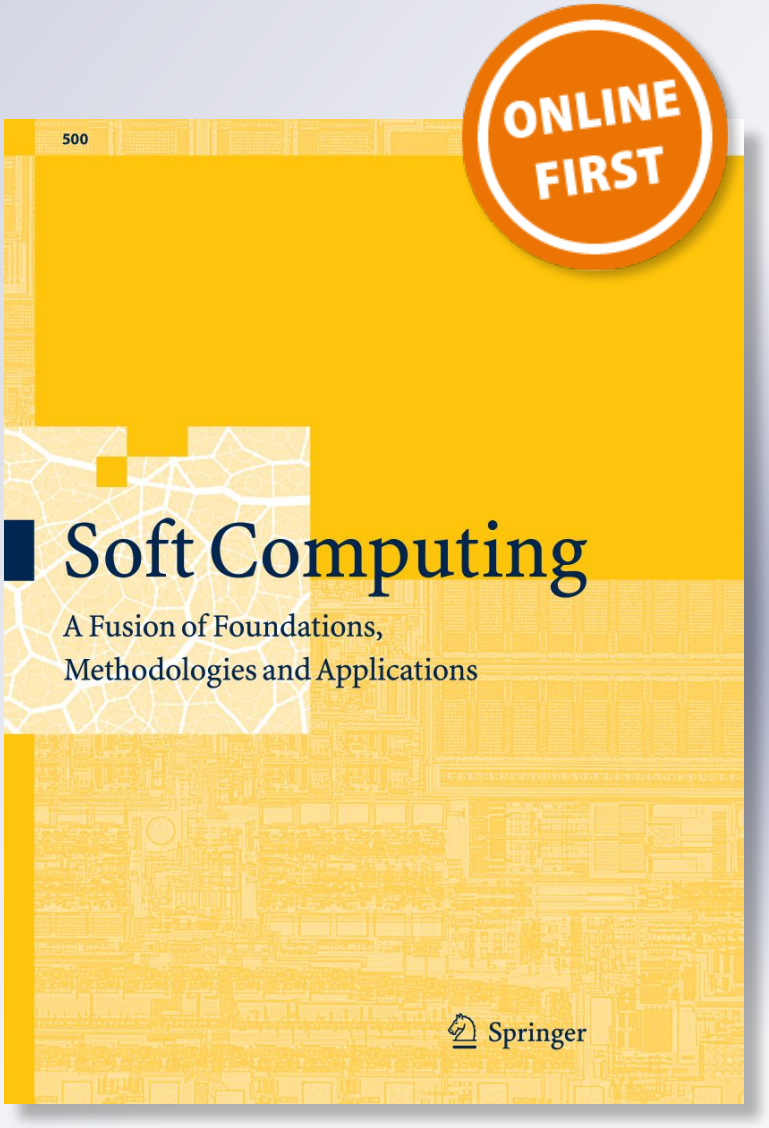
Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection

Majdi M. Mafarja & Seyedali Mirjalili

Soft Computing
A Fusion of Foundations, Methodologies and Applications

ISSN 1432-7643

Soft Comput
DOI 10.1007/s00500-018-3282-y



Springer

Soft Computing
https://doi.org/10.1007/s00500-018-3282-y

METHODOLOGIES AND APPLICATION

Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection

Majdi M. Mafarja[1] · **Seyedali Mirjalili**[2]

Abstract
Feature selection (FS) can be defined as the problem of finding the minimal number of features from an original set with the minimum information loss. Since FS problems are known as NP-hard problems, it is necessary to investigate a fast and an effective search algorithm to tackle this problem. In this paper, two incremental hill-climbing techniques (QuickReduct and CEBARKCC) are hybridized with the binary ant lion optimizer in a model called HBALO. In the proposed approach, a pool of solutions (ants) is generated randomly and then enhanced by embedding the most informative features in the dataset that are selected by the two filter feature selection models. The resultant population is then used by BALO algorithm to find the best solution. The proposed binary approaches are tested on a set of 18 well-known datasets from UCI repository and compared with the most recent related approaches. The experimental results show the superior performance of the proposed approaches in searching the feature space for optimal feature combinations.

Keywords Bio-inspired optimization · Particle swarm optimization · Binary ant lion optimizer · Approximate entropy reducts · Rough set theory · Feature selection

1 Introduction

Feature selection (FS) can be defined as the problem of searching the least subset of features which can retain a suitably high accuracy in representing the original features (Liu and Motoda 1998). FS has become a mandatory task in machine learning to eliminate the redundant and irrelevant features from the fast increasing amount of data in the real world (Jensen and Shen 2008). The existence of redundant and irrelevant features in the learning process mostly misleads learning algorithms and reduces their performance and efficiency (Theodoridis and Koutroumbas 2006). An attribute is said to be relevant if a decision depends on it, otherwise it is irrelevant. However, an attribute can be considered to be redundant if it is highly correlated with other attributes (Jensen and Shen 2002). The main objective of feature selection is to select the optimal subset by removing these features (redundant and irrelevant) and use the features that contain important information that will be obscure if any of them is excluded (Bell and Wang 2000). This area has been popular with challenging applications in operational and artificial intelligence research communities since the 1970s (Kittler 1975).

Feature selection methods can be differentiated by two main criteria: search strategy and subset evaluation (Liu and Motoda 1998). Wrappers and filters are the two most important models of feature selection when talking about the evaluation criteria (Kohavi and John 1997). The difference between these two models is that wrapper models depend on the learning algorithm used in the data mining step of the KDD process like classifiers (Wang et al. 2015), while filter models use statistical methods to eliminate features regardless of the data mining algorithm (Xue et al. 2014).

Feature selection can be seen as a search process for the optimal subset. The rapid increase in the information amount makes it impractical to use the exhaustive search in finding the optimal subset (Liu and Motoda 1998). A large number of recent works can be found in the literature trying to implement stochastic methods to tackle the challenges of feature selection problems mainly due to better local optima avoid-

Communicated by V. Loia.

✉ Majdi M. Mafarja
mmafarja@birzeit.edu

[1] Department of Computer Science, Birzeit University, PO Box 14, West Bank, Palestine

[2] Institute for Integrated and Intelligent Systems, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia

Published online: 09 June 2018

ance compared to conventional optimization techniques (Asir et al. 2016).

As the classical optimization approaches suffer from some restrictions in tackling the FS problem, meta-heuristic optimization techniques have been proposed over the last two decades in the field of feature selection as an alternative to overcome these limitations (Wang et al. 2015; Zawbaa et al. 2016). Nature-inspired algorithms (called evolutionary computation) that mimic the social and biological behavior of animals, birds, fish, wolves, etc., have shown a high performance in solving the searching problems in general (BoussaïD et al. 2013; Osman and Kelly 2012).

In the literature, many methods have been proposed in order to mimic the behavior of these species in seeking their food sources (optimal solutions). Other heuristic techniques mimic the behavior of biological and physical systems in nature. Genetic algorithm (GA) was first developed by J. Holland in the 1970s to achieve the goal of understanding the adaptive process of natural systems (Holland 1992). It is considered as the first evolution-based algorithm which has the ability to solve the nonlinear and complex problems. The traditional GA uses a population of solutions when solving a given problem; each solution is represented by a chromosome with a length of m, where m is the number of attributes in the data set (Yang and Honavar 1998). Different genetic algorithm approaches have been proposed to tackle the feature selection (Anusha and Sathiaseelan 2015; Il-Seok et al. 2004; Jensen and Shen 2004; Mafarja and Abdullah 2013; Yang and Honavar 1998). A chaotic genetic feature selection optimization method (CGFSO) is proposed in Chen et al. (2013). The main advantage of these stochastic feature selectors is that flexibility and high local optima avoidance, yet they require more function evaluation and fine tuning of parameters for dataset which is considered to be a challenge.

The particle swarm optimization algorithm (PSO), proposed by Kennedy and Eberhart (1995), is the primary swarm-based algorithm that mimics the social synergy of a flock of birds. Normally, a flock of birds or school of fish are led by a leader or 'agent'. Every single individual of the swarm follows the leading navigator based on their intuition. PSO is capable of producing quick solutions for nonlinear optimization problems (Kennedy and Eberhart 1995). Many PSO feature selection approaches can be found in Bello et al. (2007), Chakraborty et al. (2008), Wang et al. (2007), and Xue et al. (2014). Recently, PSO has been used in many FS approaches. In Moradi and Gholampour (2016), for instance, a hybrid FS method that hybridized a local search algorithm with PSO was proposed to find the salient and less correlated feature subset. An enhanced PSO was proposed in Gunasundari et al. (2016). In fact, a new variable was added to the main equation of the PSO algorithm which increased its efficiency in tackling FS problem. Other examples of using PSO for FS problem can be found in Abualigah et al. (2017).

Ant colony optimization (ACO) algorithm is another swarm-based meta-heuristic algorithm that was initially proposed by Dorigo et al. (1996). The algorithm simulates the behavior of real ants when searching for the shortest path to a food source, deposit pheromone as they travel; each ant prefers to follow the path that is rich in this pheromone. ACO mimics this pattern of behavior by applying a simple communication mechanism to enable the ant to find the shortest path between two points. An efficient ACO algorithm for feature selection has been proposed in Ke et al. (2008). For further ACO-based feature selection approaches, one can refer to Bello et al. (2005), Chen et al. (2010), Jensen and Shen (2003, 2004), and Wang et al. (2012).

Based on the biological behavior of bees, Karaboga (2005) proposed an optimization approach called the artificial bee colony (ABC) algorithm. The ABC model has three types of bees, namely employed, onlookers, and scout bees. Presumably, there is a solitary artificial employed bee for each food source. In this model, the artificial bees traverse the entire multidimensional search space. During the search, some employed and onlooker bees select food locations and change their positions based on their personal and net mates' experiences. However, some of the scout bees traverse the space in search of food without any prior experience. If a newly identified source has more nectar than the source in their memory, the bees will store the new location and discard the old location. In Shokouhifar and Sabet (2010), a hybrid approach for effective FS using neural networks and ABC optimization is described. A FS algorithm for intrusion detection systems using binary ABC is proposed in Wang et al. (2010).

Grey Wolf Optimizer (GWO) is a recent swarm intelligence algorithm (Mirjalili et al. 2014) that has been successfully employed for solving FS problems in Emary et al. (2016) and Grosan et al. (2018). In 2017, many FS that based on new optimizers were proposed in the literature including Whale Optimization Algorithm (WOA)-based FS approaches have been proposed in Mafarja and Mirjalili (2017, 2018). Other recent works in this area can be found in Mafarja et al. (2017a, b).

One of the recent algorithms is the ant lion optimizer (ALO) (Mirjalili 2015). ALO is a stochastic population-based algorithm that mimics the hunting mechanism of ant lions in nature. It has been proved that this algorithm benefits from superior performance in terms of computational time and the ability to search for the optimal or near optimal solutions (Christaline et al. 2016). An ALO-based technique has been first proposed by Zawbaa et al. (2015) to tackle the feature selection problem, in which the basic ALO was applied in a wrapper feature selection model. The proposed approach was compared with PSO- and GA-based algorithms, and the authors showed that ALO is able to delivery very competitive and promising performance.

Another FS that is based on ALO was proposed in Zawbaa et al. (2016). The proposed technique utilizes a chaotic function in adapting the single parameter that controls the balance between exploration and exploitation in the original algorithm . Later on, a binary version of the ALO algorithm was proposed by Emary et al. (2016). In this paper, two binary approaches were developed: The first approach takes only the inspiration of ALO operators and makes the corresponding binary operators. In the second approach, the native ALO is applied, while its continuous steps are thresholded using suitable threshold function after squashing them. A set of FS approaches using recent optimizers that were enhanced by employing a set of chaotic maps to control their parameters can be found in Emary and Zawbaa (2016). A recent ALO for feature selection was proposed in Mafarja et al. (2017b), where eight different transfer functions were used to convert the continuous ALO to binary to suit the FS problem. More recently, a novel feature selection approach that handled high dimensional datasets with a low number of samples using a hybrid meta-heuristic approach that hybridized GWO with ALO algorithm was proposed in Zawbaa et al. (2018). Moreover, in Emary and Zawbaa (2018), an improved FS approach using ALO algorithm was proposed. The authors improved the local search ability of ALO by employing a Levy flight random walk to help the algorithm escape from local minim.

The main question here is if there is a need for improving the algorithms proposed so far since they perform well on test cases. According to No-Free-Lunch (NFL) theorem for optimization (Wolpert 1997), there is no algorithm to solve *all* optimization problems. This mean that currently proposed algorithms for feature selection *are not able to solve all feature selection problems*. This motivated our attempt to enhance the performance of ALO to solve a wider range of problems in this field or better solves the current ones.

The main aim of this paper is tostudy the influence of the quality of the initial population on the searching progress of the ALO algorithm and on the computational time. In this approach, we proposed the use of two filter feature selection methods: Rough Set Quick Reduct (Jensen and Shen 2003) and the Conditional Entropy-Based Algorithm for Reduction of Knowledge with Computing Core (CEBARKCC) (Yu et al. 2002) to enhance the initial population. By using these two methods the most informative features will be selected to form the initial population which means that the searching process will not start from randomly generated solutions. The proposed algorithm is tested on 18 well-known datasets and shows a very good performance when compared to the other algorithms in the literature.

The rest of this paper is organized as follows: Sect. 2 discusses rough set theory and conditional entropy as evaluation and reduction mechanisms. Section 3 describes the ALO algorithm. The proposed approaches are analyzed in Sect. 4. In Sect. 5, results of our experiments are presented, and Sect. 6 addresses the conclusion and suggests future works.

2 Preliminaries

2.1 Rough set theory

Let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects called the universe of discourse; A is a non-empty set of attributes. With every attribute $a \in A$, a set of its values (V_a) is associated. For a subset of attributes $P \subseteq A$ there is an associated equivalence relation IND (P), which is called an indiscernibility relation. The relation IND (P) can be defined as follows:

$$\mathrm{IND}\,(P) = \left\{(x, y) \in U^2 | \forall a \in P, a\,(x) = a\,(y)\right\} \tag{1}$$

If $(x, y) \in \mathrm{IND}(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted$[x]p$. The indiscernibility relation is the mathematical basis of the rough set theory. In rough set theory, the lower and upper approximations are two basic operations. For a subset $X \subseteq U$. X can be approximated using only information contained within P by constructing the P-lower approximation donated as $\underline{P}\,X$, is the set of all elements of U, which can be certainly classified as elements of X based on the attribute set P. The P-upper approximation of X, denoted as $\overline{P}X$, which can be possibly classified as elements of X based on the attribute set P. These two definitions can be expressed as:

$$\underline{P}\,X = \{X | [X]_P \subseteq \underline{P}\,X\} \tag{2}$$

$$\overline{P}X = \{X | [X]_P \cap X \neq \varphi\} \tag{3}$$

Definition 1 (Dependency degree) Let $P, Q \subseteq A$, the dependency degree k is defined by:

$$k = \gamma_P(D) = \frac{\mathrm{POS}_P(Q)}{|U|} \tag{4}$$

where $|U|$ is the cardinality of U. $\mathrm{POS}_P(Q)$ called positive region, is defined by:

$$\mathrm{POS}_P\,(Q) = Y_{X \in U/Q}\,\underline{P}\,X \tag{5}$$

The positive region contains all objects of U that can be uniquely classified to blocks of the partition U/Q using the knowledge in attributes P.

For $P, Q \subset A$, it is said that Q depends on P in a degree of k $(0 \leq k \leq 1)$ denoted $P \Rightarrow_k Q$,

If $k = 1$, we say that Q depends totally on P, if $k < 1$, we claim that Q depends partially on P, and if $k = 0$, we say that Q does not depend on P.

One of the major applications of rough set theory is to find the minimal reducts by eliminating the redundant attributes from original sets, without any information loss (Pawlk 1982, 1991). The reduction of attributes can be achieved by comparing the dependency degrees of the generated subsets so that the reduced set has the same dependency degree of the original set (Jensen and Shen 2004). A reduct is formally defined as a subset R of minimal cardinality of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$ where D is a decision system.

Definition 2 (Reduct). Let R be a subset of C, R is said to be a reduct if:

$$r_R(D) = r_C(D) \forall R' \subset R, r_R(D) < r_c(D) \tag{6}$$

The intersection of all reduced subsets is called the core given in Eq. 7:

$$\text{Core}(\Re) = I_{R \in \Re} R \tag{7}$$

The core contains all those attributes that cannot be removed from the dataset without introducing more contradictions to the dataset. In the process of attribute reduction, a set $\Re_{\min} \subseteq \Re$ of reducts with minimum cardinality is searched for:

$$\Re_{min} = \{R \in \Re : |R| \leq |S|, \forall S \in \Re\} \tag{8}$$

It is obvious that finding all possible reducts is a time consuming process, and it is applicable only to small datasets as well. It is meaningless to calculate all reducts aiming to find only one minimal. To improve the performance of the above method, an alternative strategy is required for large datasets. Therefore, two important reduction methods are discussed: QuickReduct and CEBARKCC method that used the concept of rough set theory to find the minimal reducts.

2.2 Quick Reduct

The QuickReduct algorithm given in Algorithm 1 and obtained from Jensen and Shen (2003) starts with an empty set and adds the best candidate attribute that increases the dependency degree until the consistent state (1 if the dataset is consistent) is achieved. QuickReduct algorithm attempts to find the minimal reduct without exhaustively generating all possible solutions which will be computationally expensive.

Table 1 Example dataset

$x \in U$	a	b	c	d	$\Rightarrow e$
$u0$	1	0	2	2	0
$u1$	0	1	1	1	2
$u2$	2	0	0	1	1
$u3$	1	1	0	2	2
$u4$	1	0	2	0	1
$u5$	2	2	0	1	1
$u6$	2	1	1	1	2
$u7$	0	1	1	0	1

Algorithm 1. QuickReduct (C, D)
C, the set if all conditional features;
D, the set of decision features;
(1) $R \longleftarrow \{\ \}$
(2) do
(3) $\quad T \longleftarrow R$
(4) $\quad \forall x \in (C - R)$
(5) $\quad\quad$ if $\gamma_{R \cup \{x\}} > \gamma_T(D)$
(6) $\quad\quad\quad T \longleftarrow R \cup \{x\}$
(7) $\quad R \longleftarrow T$
(8) until $\gamma_R(D) == \gamma_C(D)$
(9) return R

In the example represented in Table 1, the attribute d is initially chosen as its corresponding degree of dependency is the highest (a value of 0.25). Next, the subsets $\{a, d\}$, $\{b, d\}$, and $\{c, d\}$ are evaluated. The subset {b, d} produces a dependency degree of 1 and the algorithm terminates as a reduct has been found. The generated reduct shows the way of reducing the dimensionality of the original dataset by eliminating those conditional attributes that do not appear in the set.

By using QuickReduct algorithm to enhance the initial population, we make sure that all solutions are close to minimal reducts and contain the most informative features that may lead to higher accuracy in the following stages.

2.3 Approximate entropy reducts

2.3.1 Conditional information entropy

Rough set entropy reduct (Ślezak 2002) is the reduction based on conditional information entropy. For a decision system $I = (U, A \cup \{d\})$, A is the condition attributes and d is decision attribute. B is a subset of attributes where

$B \subseteq A$. The entropy of B is defined as in Eq. 9.

$$H(B) = -\sum_{X_i \in U/B} p(X_i) \log(p(X_i)) \tag{9}$$

If $U/d = \{Y_1, Y_2, \ldots, Y_m\}$, the conditional entropy of B with reference to d is as in Eq. 10:

$$H(d|B) = -\sum_{i=1}^{n} p(X_i) \sum_{j=1}^{m} p\left(Y_j|X_i\right) \log p\left(Y_j|X_i\right) \tag{10}$$

where $p(Y_j|X_i) = \frac{Y_j \cap X_i}{|X_i|}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. The rough set entropy reduct can be defined as in Eq. 11:

$$\begin{aligned} \text{Reduct} = \{R \subseteq A | H(d|R) = H(d|A), \\ \forall B \subset R H(d|B) \neq H(d|A)\} \end{aligned} \tag{11}$$

2.3.2 Approximate entropy reducts

The approximate entropy reduct (Ślezak 2002) is defined as in Eq. 12:

$$H(d|B) + \log(1 - \varepsilon) \leq H(d|A) \tag{12}$$

where $\varepsilon \in [0, 1)$, $B \subseteq A$.

The approximate reducts is the attribute subsets that can approximate decision classes accurately enough based on the slack factor ε. The minimal ε-approximate entropy reducts is to find a minimal B satisfying Eq. 12, which is NP-hard for any $\varepsilon \in [0, 1)$.

2.4 The CEBARKCC method

Another technique for discovering rough set reducts is entropy-based reduction called CEBARKCC developed from work carried out in Yu et al. (2002) represented in Algorithm 2. This approach used the forward greedy search strategy to search a distribution reduct where $H(D|B \cup \{a_i\}$ is used as the heuristic information.

In CEBARKCC the reduction process is being processed step by step by adding one attribute at a time to the reduct and the conditional entropy is calculated, if an attribute a cannot add any information to the classes induced by an attributes subset B, namely $H(D|B \cup \{ai\}) = H(D|B)$, it may be reduced (Yu et al. 2002). The algorithm stopped when the condition $H(D|C) = H(D|B)$ is satisfied.

Algorithm 2. CEBARKCC (conditional entropy-based algorithm for reduction of knowledge with computing core):

Input: A decision table $I = (U, A \cup \{d\})$
Output: A relative reduction $B \subseteq C$.
Step1: Calculate $H(D|C)$;
Step2: Calculate the core $C0$ and $Att := C - C0$;
Step3: Let B: $=C0$,
(1) If $|B|! = 0$, then
calculate $H(D|B)$ and go to (4);
Else go to (2).
(2) For every $a_i \in Att$
calculate $H(D|\ B \cup \{ai\})$.
(3) Set aj: $= \min\{H(D|\ B \cup \{ai\}), ai \in Att\}$,
set Att :$=Att$-$\{aj\}$ and B:$=B \cup \{aj\}$;
(4) If $H(D|B)$=$H(D|C)$, then
stop;
Else go to (2);

3 Ant Lion optimizer (ALO)

ALO is a nature-inspired algorithm that mimics the foraging behavior of the ant lions' larvae. ALO was proposed by Mirjalili (2015). The ALO algorithm is a stochastic population-based optimization algorithm that can be considered as one of the meta-heuristics (Mirjalili 2015) that has a very good performance in tackling a wide range of optimization problems.

Ant lions' life is divided into two stages: the larva stage and adulthood. The most important characteristic of the larva stage is hunting. The ant lion larvae digs a trap in the sand which takes the form of a cone by moving in a circular motion and throws sand outside the trap the area using its massive jaws. The size of the trap depends on two factors; the hunger level and the moon shape (Goodenough et al. 2009; Hutchins and Olendorf 2004; Mirjalili 2015). After digging a trap, the larvae disappears under the bottom of the cone and waits for its preferable prey (ants) to be trapped in the hole. Once the ant lion realizes that the prey is in the trap, it tries to catch it. However, insects usually try to escape from the cone and are not caught immediately. In this case, the ant lion throws sand intelligently toward the edge of the trap until the prey slides toward the bottom of the hole again. When the prey slips to the bottom of the hole and is caught by the jaw, it is pulled under the soil and consumed. After the consumption of prey, ant lions throw food scraps out of the hole and adjust the hole to hunt other prey. Based on this description of the ant lion hunting process, a set of conditions can be formulated as in the following items (Mirjalili 2015):

- Preys (ants) move randomly around the search space. These moves are affected by the traps of ant lions.
- The highest fitness ant lion builds a larger pit.
- Catching an ant by an ant lion is proportional to the fitness of that ant lion.

- Each ant lion can catch an ant in each iteration.
- To simulate sliding ants toward ant lions, the range of random walks is decreased adaptively.
- If an ant becomes fitter than an ant lion, this means that it is caught and pulled under the soil by the ant lion.
- An ant lion repositions itself to the latest caught prey and builds a pit to improve its chances of catching another prey after each hunt.

3.1 Random walks of ants

Ants move around the search space (update their positions) using random walks at each iteration of the algorithm based on Eq. 13.

$$X(t) = [0, \text{cumsum}(2r(t1) - 1), \text{cumsum}(2r(t1) - 1), \ldots, \text{cumsum}(2r(tn) - 1)] \quad (13)$$

where cumsum represents the cumulative sum, n is the max iteration, t is the iteration, and $r(t)$ is a stochastic function that takes value (1) if a random number is less than 0.5 and 0 otherwise.

To insure that the ants move within the boundaries of the search space, the random walks are normalized using Eq. 14 (min–max normalization) in each iteration:

$$X_i^t = \frac{\left(X_i^t - a_i\right) \times \left(d_i - c_i^t\right)}{\left(d_i^t - a_i\right)} + c_i \quad (14)$$

where a_i and b_i are the minimum and maximum of random walk of ith variable, c_i^t and d_i^t are the minimum and maximum ith variable in tth iteration.

3.2 Trapping in ant lion's pits

Random walks of ants are affected by the traps of the ant lions. Equations 15 and 16 model this assumption:

$$c_i^t = Ant\ lion_j^t + c^t \quad (15)$$

$$d_i^t = Ant\ lion_j^t + d^t \quad (16)$$

where c^t and d^t are two vectors that contain the minimum and maximum of all variables in tth iteration, c_i^t and d_i^t are the minimum and maximum ith ant and $Ant\ lion_j^t$ represents the position of the jth ant lion at the tth iteration.

3.3 Building a trap

A selection mechanism should be used to model the hunting capability of ant lions. The ant lion with the higher fitness has a higher chance to catch an ant. In this work, roulette wheel selection (RWS) for selecting ant lions based on their fitness value was applied.

3.4 Sliding ants toward ant lion

When the ant slips into the pit, it tries to escape. When the ant lion realizes that there is a prey in the pit it shoots the sand outwards the center of the pit. To model this behavior mathematically, the radius of the ant's random walk is decreased adaptively using Eqs. 17 and 18.

$$c^t = \frac{c^t}{I} \quad (17)$$

$$d^t = \frac{d^t}{I} \quad (18)$$

where c^t and d^t are vectors that represent the minimum and maximum of all variables at tth iteration, I is a ration, which is defined in Eq. 19.

$$I = 10^w \frac{t}{T} \quad (19)$$

where t is the current iteration, T is the max iteration, and w is a constant that can adjust the accuracy level of the exploitation. w is 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 $t > 0.95T$).

3.5 Catching prey and rebuilding the pit

In the final stage of hunting, the prey reaches the bottom of the pit and is caught in the ant lion's jaw. After that, the ant lion pulls the ant inside the sand and consumes its body. It is assumed that prey is caught when the ant becomes fitter than its corresponding ant lion. Here, the ant lion has to update its position to the position of the hunted ant to increase its ability of hunting new ant. Equation 20 models this process (for a minimization problem without the loss of generality):

$$Ant\ lion_j^t = \begin{cases} Ant_i^t & \text{if } f\left(Ant_i^t\right) < f\left(Ant\ lion_j^t\right) \\ Ant\ lion_j^t & \text{otherwise} \end{cases} \quad (20)$$

where t represents the current iteration, $Ant\ lion_j^t$, Ant_i^t represent the position of the jth ant lion and the ith ant at the tth iteration.

3.6 Elitism

The ant lion with the higher fitness in each iteration is considered as an elite. The elite ant lion and the selected ant lion

by using the selection mechanism guide the random walk of an ant, and hence the repositioning of a given ant follows the following Eq. 9.

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{21}$$

where R_A^t represents the random walk around the selected ant lion using the selection mechanism, and R_E^trepresents the random walk around the elite ant lion.

4 Hybrid binary ant lion optimizer for feature selection

In this section, the two proposed hybrid approaches are exploited in feature selection for classification problems. The proposed ant lion Optimizer is mainly based on the hybridization between rough set and conditional entropy-based approaches (QuickReduct and CEBARKCC) and ALO-based approaches namely (BALO-1, BALO-S, and BALO-V)) reported in Emary et al. (2016). These three approaches are ALO-based approaches where the solution is represented in a binary format instead of using the continuous version in cALO (Zawbaa et al. 2016). The average operator in the ALO is replaced by crossover operation between two binary solutions. The two solutions to perform the crossover are either obtained by performing mutation as a local search around ant lions with suitable mutation rate, called BALO-1, or as a threshold continuous random walk around ant lions with suitable step size, called BALO-S and BALO-V.

For further information about these approaches, readers are referred to Zawbaa et al. (2016) and Emary et al. (2016).

As shown in Fig. 1, in the HBALO, both QuickReduct and CEBARKCC are employed to search for the minimal reducts. An initial population for the ants is randomly generated, and two solutions (reducts) are generated for each data set using QuickReduct and CEBARKCC approaches, then the ants' population is updated by applying the OR logical operator between each solution in the population and the redcut. This step will include the most informative attributes in each solution of the population. The algorithm then starts from a population that contains the most informative attributes in each individual of the population. In order to balance between the number of selected features and the quality of each reduct, the fitness function in Eq. 22 is used (Jensen and Shen 2003).

$$\text{Fitness}\,(R) = \gamma_R\,(D) * \frac{|C| - |R|}{|C|}, \tag{22}$$

where $\gamma_R\,(D)$ represents the dependency degree in the QuickReduct and the conditional entropy for CEBARKCC. $|R|$ is the cardinality of the selected subset and $|C|$ is the total number of features in the dataset.

The fitness function that is used in BALO approaches to evaluate individual search agents is shown in Eq. 23 (Emary et al. 2016).

$$Fitness = \alpha\gamma_R(D) + \beta\frac{R}{C} \tag{23}$$

where $\gamma_R(D)$ represents the classification error rate of a given classier (the K-nearest neighbor (KNN) classifier (Chuang et al. 2008) 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 $\beta = (1 - \alpha)$ adopted from Emary et al. (2016). The source code of the proposed method can be found in http://www.alimirjalili.com/Projects.html

5 Experimental results and discussion

The implementation of the proposed algorithms is done in MATLAB. The performance of the proposed algorithms is tested using eighteen FS benchmark datasets in Table 2 from the UCI data repository (Blake and Merz 1998). The selected datasets contain various number of features and instances; some of them are of high dimensionality which insures performance of optimization algorithms in huge search spaces.

A filter-wrapper approach for feature selection is used. First, two filter approaches based on rough set and conditional entropy are used to generate the initial population and then the wrapper approach based on the KNN classifier [where $K = 5$ (Emary et al. 2016)] is used to generate the best reduct. In the wrapper approach, the training/testing model is used, where 80% of each individual dataset is used to build the learning model, while the other 20% of the instance is used for testing purposes (Friedman et al. 2001).

The experiments are tested on Intel machine Core i5 CPU 2.2 GHz and 4 GB RAM, and the parameter values are in Table 3 as adopted from Emary et al. (2016).

The proposed approaches are tested and compared against the ALO optimizer and other three enhanced ALO optimizers namely BALO-1, BALO-S, and BALO-V (Emary et al. 2016) where each of them used three different initialization methods in generating the ant lion positions; namely *Mixed initialization*, *small initialization*, and *large initialization*.

In the fitness function, two objectives are adopted. The first one is the classification accuracy while the second is the number of selected features in each reduct. The performance of the QuickReduct-based approaches over the first objective (classification accuracy) is outlined in Tables 4, 5 and 6.

From Tables 4, 5 and 6, it is evident that the classification accuracy produced when using the full feature set is worse than using the three proposed approaches. In addition, bALO-

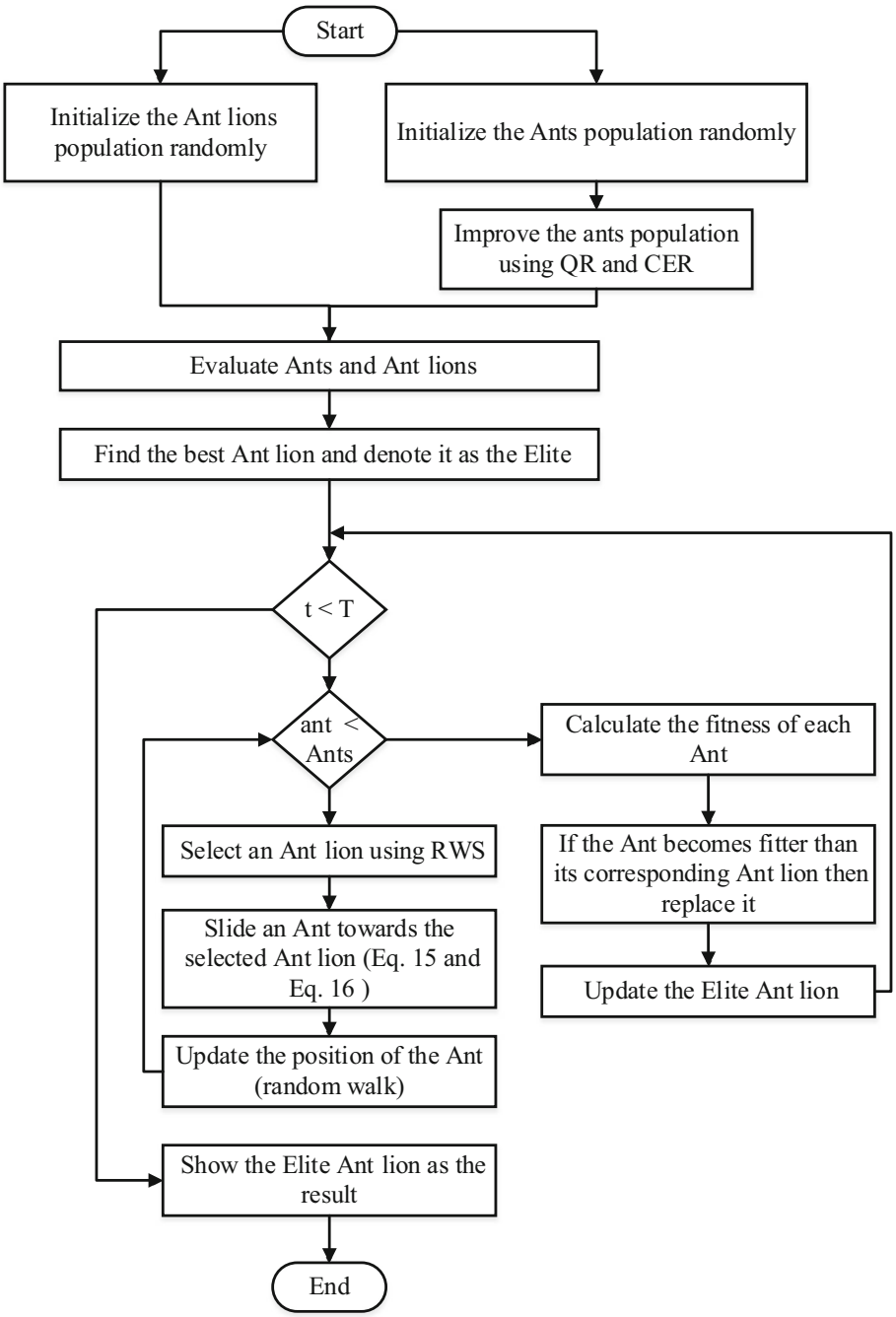


Fig. 1 Flowchart of the proposed ant lion optimizer (HBALO)

QR that based on the basic ALO algorithms outperforms all other approaches on five datasets when compared when using large and small-mixed initialization, and on 6 datasets when using small initialization. bALOV-QR approach performs better than other approaches on eight datasets regardless of the initialization method used.

In Emary et al. (2016), it was stated that the mixed initialization-based methods outperform other approaches. From the previous results, we can see that the QuickReduct-based approaches are able to produce better results on many datasets than those produced by the mixed initialization-based methods. The performance of the proposed approach can be interpreted by the assumption that the generated population contains the closest subsets to the optimal solution. This mechanism enhanced the ability to concentrate on the classification accuracy in the fitness function since the solutions are closed to the minimal.

Tables 7, 8, and 9 represent the comparison between the CEBARKCC-based approaches and the other approaches that use the large, small and mixed-small initializations, respectively. Both bALO-CE and bALOV-CE outperformed

Table 2 List of datasets used in the experiments

	Dataset	No. of features	No. of samples
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

Table 3 Parameter setting for experiments

Parameter	Value(s)
No. of search agents	8
No. of iterations	70
Problem dimension	Number of features in the data
Search domain in binary algorithms	{0, 1}
Number of runs	20
α parameter in the fitness function	0.99
β parameter in the fitness function	0.01

other approaches over five datasets with a significant difference (reached 13%) in some cases.

Although the QuickReduct-based approaches perform better than CEBARKCC in terms of classification accuracy, the latter, still has a good performance when compared against other approaches. This might be because QuickReduct simulates the forward selection mechanism when generating the next subset where the algorithm starts from an empty set and adds only the feature that improves the quality of the solution.

Tables 10 and 11 outline the secondary objective in the fitness function namely average selection size. bALOV-CE has a much enhanced performance over the other optimizers adopted in the paper where it outperformed all other approaches on seven datasets. Other approaches have shown good performance over many datasets. The QuickReduct-based approaches showed a good performance in terms of the number of selected features but the CEBARKCC-based approaches still perform better, while the former outperforms the latter in terms of classification accuracy. This is not contradictory, since the two methods used to reduce the number of the selected features are independent from the classification algorithm in a filter mode. This proves that the selected features by the QuickReduct algorithm are the most informative features in the dataset.

Figures 2 and 3 shows the representation of the comparison between QuickReduct and CEBARKCC based approaches respectively, and other approaches from literature in terms of selection size, where our approaches are the first three approaches in each figure. It can be seen that the proposed approaches outperform other approaches on many datasets.

In Tables 12 and 13 the average computational time of different optimization algorithms is outlined. Since all approaches are using the same parameter settings, we can use the computational time to compare the performance of the algorithms. Inspecting Table 12, it may be seen that bALO-QR bALO-CE (bALO1 based) have the minimum running time among other approaches. From these observations we can remark that the bALO1-based approaches are able to produce the results faster than other approaches due to its simple operators that depend on simple mutation and crossover. In addition, there is a significant difference between the performance of BALOS- and BALOV-based approaches and all other adopted approaches in this paper. This proves the convergence capability of these approaches and their ability to provide better results in terms of classification accuracy and selection ratio in a very short time. Figures 4 and 5 visualize the comparison between the proposed approaches and the other obtained approaches from literature in terms of computational time.

Taken together, the results of this section show that the initial population influences the robustness and the convergence of ALO algorithm. In the proposed algorithms the initial population is generated in two phases. In the first phase, a simple method based on the mathematical theory of random number generation is employed to generate a sequence of random numbers by spreading points uniformly in the search space. Then, in the second phase, the two incremental local search techniques (QuickReduct and CEBARKCC) are used to enhance the random population by embedding the most informative features with the randomly generated solutions. The first phase ensures diversity of the population, while the second tries to converge the initial solutions toward the optimal solution. Here, we can remark that the enhanced performance of the proposed algorithms is due to the diversity in population provided by this method and the closeness of some search agents to the global optima. It is worth mention-

Table 4 Comparison of classification accuracy between QuickReduct-based approaches and approaches that use large initialization

DS no.	Dataset	bALO-QR	bALOS-QR	bALOV-QR	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	0.974	0.945	0.974	0.969	0.974	0.966	**0.978**	0.944
2	BreastEW	0.962	0.916	0.961	0.972	0.977	0.976	**0.979**	0.963
3	CongressEW	0.972	0.916	**0.973**	0.960	0.964	0.956	0.968	0.917
4	Exactly	0.912	0.640	**0.974**	0.705	0.777	0.746	0.857	0.673
5	Exactly2	0.760	0.702	0.758	0.766	0.767	0.768	**0.771**	0.743
6	HeartEW	0.884	0.804	**0.889**	0.856	0.876	0.867	0.876	0.815
7	IonosphereEW	0.869	0.795	0.884	0.897	**0.903**	0.885	0.893	0.866
8	KrvskpEW	**0.975**	0.761	0.975	0.941	0.965	0.946	0.967	0.915
9	Lymphography	**0.886**	0.711	0.878	0.830	0.845	0.827	0.865	0.683
10	M-of-n	**1.000**	0.701	**1.000**	0.901	0.989	0.898	0.980	0.849
11	PenglungEW	0.665	0.573	0.659	0.962	0.964	0.958	**0.966**	0.951
12	SonarEW	0.840	0.717	0.852	0.832	**0.853**	0.834	0.844	0.620
13	SpectEW	**0.900**	0.788	0.894	0.858	0.886	0.863	0.886	0.831
14	Tic-tac-toe	0.800	0.654	**0.811**	0.772	0.787	0.787	0.783	0.715
15	Vote	0.948	0.887	0.953	0.957	0.960	0.953	**0.963**	0.877
16	WaveformEW	0.794	0.689	0.798	0.785	0.797	0.781	**0.800**	0.768
17	WineEW	**1.000**	0.921	**1.000**	0.983	0.992	0.992	0.992	0.932
18	Zoo	0.961	0.816	**0.973**	0.906	0.921	0.916	0.911	0.792

Bold values indicate the best results

Table 5 Comparison of classification accuracy between QuickReduct-based approaches and approaches that use small initialization

DS no.	Dataset	bALO-QR	bALOS-QR	bALOV-QR	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	**0.974**	0.945	**0.974**	0.967	0.974	0.967	0.938	0.944
2	BreastEW	0.962	0.916	0.961	0.976	0.976	**0.979**	0.962	0.963
3	CongressEW	0.972	0.916	**0.973**	0.959	0.966	0.960	0.941	0.917
4	Exactly	0.912	0.640	**0.974**	0.699	0.861	0.730	0.718	0.673
5	Exactly2	0.760	0.702	0.758	0.762	**0.768**	0.767	0.748	0.743
6	HeartEW	0.884	0.804	**0.889**	0.857	0.870	0.870	0.802	0.815
7	IonosphereEW	0.869	0.795	0.884	**0.900**	0.887	0.885	0.832	0.866
8	KrvskpEW	**0.975**	0.761	0.975	0.955	0.963	0.948	0.962	0.915
9	Lymphography	**0.886**	0.711	0.878	0.827	0.862	0.848	0.776	0.683
10	M-of-n	**1.000**	0.701	**1.000**	0.909	0.980	0.903	0.945	0.849
11	PenglungEW	0.665	0.573	0.659	0.961	**0.963**	0.958	0.955	0.951
12	SonarEW	0.840	0.717	0.852	0.825	**0.858**	0.829	0.702	0.620
13	SpectEW	**0.900**	0.788	0.894	0.846	0.884	0.863	0.807	0.831
14	Tic-tac-toe	0.800	0.654	**0.811**	0.764	0.787	0.779	0.756	0.715
15	Vote	0.948	0.887	0.953	0.948	**0.962**	0.952	0.918	0.877
16	WaveformEW	0.794	0.689	0.798	0.792	**0.800**	0.776	0.781	0.768
17	WineEW	**1.000**	0.921	**1.000**	0.992	0.994	0.989	0.921	0.932
18	Zoo	0.961	0.816	**0.973**	0.906	0.921	0.921	0.831	0.792

Bold values indicate the best results

ing here that the proposed initialization methods enhanced the ability of ALO algorithm balance between exploration and exploitation.

Also, the results of this section showed the superior performance of the proposed HBALO as compared to the current algorithms in the literature. The ALO algorithm benefits from high exploration due to the position updating equations. The solutions constantly face random changes using multiple best solutions obtained so far. The search landscape of feature selection problems changes for every dataset. Also, such

Table 6 Comparison of classification accuracy between QuickReduct-based approaches and approaches that use small mixed initialization

DS no.	Dataset	bALO-QR	bALOS-QR	bALOV-QR	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	**0.974**	0.945	**0.974**	0.971	**0.974**	0.969	0.970	0.944
2	BreastEW	0.962	0.916	0.961	0.972	**0.979**	**0.979**	**0.979**	0.963
3	CongressEW	0.972	0.916	**0.973**	0.961	0.970	0.961	0.963	0.917
4	Exactly	0.912	0.640	**0.974**	0.701	0.856	0.723	0.856	0.673
5	Exactly2	0.760	0.702	0.758	0.764	**0.766**	**0.766**	**0.766**	0.743
6	HeartEW	0.884	0.804	**0.889**	0.869	0.872	0.867	0.878	0.815
7	IonosphereEW	0.869	0.795	0.884	0.885	0.889	0.877	**0.892**	0.866
8	KrvskpEW	**0.975**	0.761	0.975	0.948	0.967	0.946	0.966	0.915
9	Lymphography	**0.886**	0.711	0.878	0.824	0.875	0.844	0.861	0.683
10	M-of-n	**1.000**	0.701	**1.000**	0.930	0.994	0.917	0.990	0.849
11	PenglungEW	0.665	0.573	0.659	0.962	0.963	0.959	**0.964**	0.951
12	SonarEW	0.840	0.717	0.852	0.827	**0.868**	0.825	0.846	0.620
13	SpectEW	**0.900**	0.788	0.894	0.850	0.890	0.863	0.891	0.831
14	Tic-tac-toe	0.800	0.654	**0.811**	0.772	0.787	0.779	0.787	0.715
15	Vote	0.948	0.887	0.953	0.950	0.955	0.953	**0.960**	0.877
16	WaveformEW	0.794	0.689	0.798	0.786	0.800	0.778	**0.805**	0.768
17	WineEW	**1.000**	0.921	**1.000**	0.989	0.989	0.989	0.994	0.932
18	Zoo	0.961	0.816	**0.973**	0.891	0.931	0.906	0.921	0.792

Bold values indicate the best results

Table 7 Comparison of classification accuracy between CEBARKCC-based approaches and approaches that use large initialization

DS no.	Dataset	bALO-CE	bALOS-CE	bALOV-CE	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	0.974	0.955	0.974	0.969	0.974	0.966	**0.978**	0.944
2	BreastEW	0.968	0.927	0.962	0.972	0.977	0.976	**0.979**	0.963
3	CongressEW	**0.970**	0.881	0.965	0.960	0.964	0.956	0.968	0.917
4	Exactly	0.962	0.653	**0.980**	0.705	0.777	0.746	0.857	0.673
5	Exactly2	0.763	0.718	0.762	0.766	0.767	0.768	**0.771**	0.743
6	HeartEW	0.859	0.776	0.868	0.856	**0.876**	0.867	**0.876**	0.815
7	IonosphereEW	0.890	0.815	0.892	0.897	**0.903**	0.885	0.893	0.866
8	KrvskpEW	**0.973**	0.718	0.972	0.941	0.965	0.946	0.967	0.915
9	Lymphography	0.902	0.624	**0.904**	0.830	0.845	0.827	0.865	0.683
10	M-of-n	**1.000**	0.719	0.994	0.901	0.989	0.898	0.980	0.849
11	PenglungEW	0.881	0.784	0.872	0.962	0.964	0.958	**0.966**	0.951
12	SonarEW	0.869	0.754	**0.906**	0.832	0.853	0.834	0.844	0.620
13	SpectEW	**0.888**	0.813	0.882	0.858	0.886	0.863	0.886	0.831
14	Tic-tac-toe	**0.820**	0.695	0.815	0.772	0.787	0.787	0.783	0.715
15	Vote	0.952	0.859	0.955	0.957	0.960	0.953	**0.963**	0.877
16	WaveformEW	0.807	0.681	**0.812**	0.785	0.797	0.781	0.800	0.768
17	WineEW	0.987	0.987	0.984	0.983	**0.992**	**0.992**	**0.992**	0.932
18	Zoo	0.957	0.847	**0.961**	0.906	0.921	0.916	0.911	0.792

Bold values indicate the best results

problems have higher number of variables due to their binary nature. To handle such challenges, an algorithm requires high exploratory behavior to avoid the local optimal solutions and find the global optimum. The main issue with promoting exploration is the reduction in the accuracy of solutions. This was the motivation of integrating hill-climbing technique into ALO and the proposal of HBALO. Hill climbing is one of the best local search techniques in the literature. This algorithm improves the accuracy of the solutions obtained by the ALO algorithm in the proposed method. The results showed that

Table 8 Comparison of classification accuracy between CEBARKCC-based approaches and approaches that use small initialization

DS no.	Dataset	bALO-CE	bALOS-CE	bALOV-CE	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	**0.974**	0.955	**0.974**	0.967	0.974	0.967	0.938	0.944
2	BreastEW	0.968	0.927	0.962	0.976	0.976	**0.979**	0.962	0.963
3	CongressEW	**0.970**	0.881	0.965	0.959	0.966	0.960	0.941	0.917
4	Exactly	0.962	0.653	**0.980**	0.699	0.861	0.730	0.718	0.673
5	Exactly2	0.763	0.718	0.762	0.762	**0.768**	0.767	0.748	0.743
6	HeartEW	0.859	0.776	0.868	0.857	**0.870**	**0.870**	0.802	0.815
7	IonosphereEW	0.890	0.815	0.892	**0.900**	0.887	0.885	0.832	0.866
8	KrvskpEW	**0.973**	0.718	0.972	0.955	0.963	0.948	0.962	0.915
9	Lymphography	0.902	0.624	**0.904**	0.827	0.862	0.848	0.776	0.683
10	M-of-n	**1.000**	0.719	0.994	0.909	0.980	0.903	0.945	0.849
11	PenglungEW	0.881	0.784	0.872	0.961	**0.963**	0.958	0.955	0.951
12	SonarEW	0.869	0.754	**0.906**	0.825	0.858	0.829	0.702	0.620
13	SpectEW	**0.888**	0.813	0.882	0.846	0.884	0.863	0.807	0.831
14	Tic-tac-toe	**0.820**	0.695	0.815	0.764	0.787	0.779	0.756	0.715
15	Vote	0.952	0.859	0.955	0.948	**0.962**	0.952	0.918	0.877
16	WaveformEW	0.807	0.681	**0.812**	0.792	0.800	0.776	0.781	0.768
17	WineEW	0.987	0.987	0.984	0.992	**0.994**	0.989	0.921	0.932
18	Zoo	0.957	0.847	**0.961**	0.906	0.921	0.921	0.831	0.792

Bold values indicate the best results

Table 9 Comparison of classification accuracy between CEBARKCC-based approaches and approaches that use small-mixed initialization

DS no.	Dataset	bALO-CE	bALOS-CE	bALOV-CE	ALO	BALO-1	BALO-S	BALO-V	Full
1	Breastcancer	**0.974**	0.955	**0.974**	0.971	**0.974**	0.969	0.970	0.944
2	BreastEW	0.968	0.927	0.962	0.972	**0.979**	**0.979**	**0.979**	0.963
3	CongressEW	0.970	0.881	0.965	0.961	**0.970**	0.961	0.963	0.917
4	Exactly	0.962	0.653	**0.980**	0.701	0.856	0.723	0.856	0.673
5	Exactly2	0.763	0.718	0.762	0.764	**0.766**	**0.766**	**0.766**	0.743
6	HeartEW	0.859	0.776	0.868	0.869	0.872	0.867	**0.878**	0.815
7	IonosphereEW	0.890	0.815	0.892	0.885	0.889	0.877	0.892	0.866
8	KrvskpEW	**0.973**	0.718	0.972	0.948	0.967	0.946	0.966	0.915
9	Lymphography	0.902	0.624	**0.904**	0.824	0.875	0.844	0.861	0.683
10	M-of-n	**1.000**	0.719	0.994	0.930	0.994	0.917	0.990	0.849
11	PenglungEW	0.881	0.784	0.872	0.962	0.963	0.959	**0.964**	0.951
12	SonarEW	0.869	0.754	**0.906**	0.827	0.868	0.825	0.846	0.620
13	SpectEW	0.888	0.813	0.882	0.850	0.890	0.863	**0.891**	0.831
14	Tic-tac-toe	**0.820**	0.695	0.815	0.772	0.787	0.779	0.787	0.715
15	Vote	0.952	0.859	0.955	0.950	0.955	0.953	**0.960**	0.877
16	WaveformEW	0.807	0.681	**0.812**	0.786	0.800	0.778	0.805	0.768
17	WineEW	0.987	0.987	0.984	0.989	0.989	0.989	**0.994**	0.932
18	Zoo	0.957	0.847	**0.961**	0.891	0.931	0.906	0.921	0.792

Bold values indicate the best results

the high exploratory mechanism of ALO and accurate local search of hill climbing combined are able to provide superior results when solving feature selection problems.

The comparative results of HBALO with BALO-S and BALO-V showed the superior performance of the proposed algorithm as well. The BALO-S and BALO-V algorithms uses different transfer functions. The literature shows that the performance of binary algorithms can be significantly improved with changing the transfer function. However, such transfer functions change the way that an algorithm flips the

Table 10 Comparison between QuickReduct-based approaches and other optimizers on all the data sets averaged on all initialization methods

DS no.	Dataset	bALO-QR	bALOS-QR	bALOV-QR	ALO	BALO-1	BALO-S	BALO-V
1	Breastcancer	0.444	0.444	0.444	0.519	**0.419**	0.513	0.443
2	BreastEW	0.447	0.547	**0.440**	0.519	0.494	0.537	0.537
3	CongressEW	0.275	0.313	0.288	**0.271**	0.313	0.399	0.309
4	Exactly	0.462	0.585	0.462	**0.406**	0.436	0.517	0.462
5	Exactly2	0.446	0.446	0.415	0.457	0.338	0.547	**0.333**
6	HeartEW	0.462	0.646	**0.446**	0.650	0.474	0.573	0.504
7	IonosphereEW	0.388	**0.294**	0.335	0.335	0.431	0.350	0.399
8	KrvskpEW	**0.389**	0.522	0.411	0.688	0.457	0.559	0.468
9	Lymphography	0.500	0.578	0.478	0.417	**0.361**	0.475	0.432
10	M-of-n	**0.462**	0.615	**0.462**	0.684	0.483	0.675	0.513
11	PenglungEW	0.424	0.497	0.396	**0.111**	0.393	0.329	0.374
12	SonarEW	0.403	0.457	0.423	**0.286**	0.443	0.444	0.462
13	SpectEW	0.345	0.391	**0.327**	0.508	0.391	0.543	0.389
14	Tic-tac-toe	0.733	0.644	0.733	0.691	0.654	**0.617**	0.630
15	Vote	0.350	0.413	0.388	**0.285**	0.326	0.292	0.330
16	WaveformEW	0.550	0.650	**0.525**	0.847	0.593	0.654	0.567
17	WineEW	0.431	0.569	0.400	0.526	0.444	0.470	**0.389**
18	Zoo	0.375	0.463	0.413	0.441	**0.361**	0.458	0.396

Bold values indicate the best results

Table 11 Comparison between CEBARKCC-based approaches and other optimizers on all the data sets averaged on all initialization methods

DS no.	Datset	*Atts*	*Objects*	bALO-CE	bALOS-CE	bALOV-CE	ALO	BALO-1	BALO-S	BALO-V
1	Breastcancer	9	699	0.400	0.422	**0.333**	0.519	0.419	0.513	0.443
2	BreastEW	30	569	0.480	0.613	**0.400**	0.519	0.494	0.537	0.537
3	CongressEW	16	435	0.350	0.375	0.338	**0.271**	0.313	0.399	0.309
4	Exactly	13	1000	0.462	0.585	0.462	**0.406**	0.436	0.517	0.462
5	Exactly2	13	1000	0.138	0.308	**0.077**	0.457	0.338	0.547	0.333
6	HeartEW	13	270	0.523	0.615	0.615	0.650	**0.474**	0.573	0.504
7	IonosphereEW	34	351	0.318	0.294	**0.271**	0.335	0.431	0.350	0.399
8	KrvskpEW	36	3196	**0.433**	0.589	0.439	0.688	0.457	0.559	0.468
9	Lymphography	18	148	**0.356**	0.422	0.367	0.417	0.361	0.475	0.432
10	M-of-n	13	1000	**0.462**	0.585	**0.462**	0.684	0.483	0.675	0.513
11	PenglungEW	325	73	0.402	0.462	0.414	**0.111**	0.393	0.329	0.374
12	SonarEW	60	208	0.423	0.513	0.450	**0.286**	0.443	0.444	0.462
13	SpectEW	22	267	**0.327**	0.409	0.373	0.508	0.391	0.543	0.389
14	Tic-tac-toe	9	958	**0.556**	0.667	**0.556**	0.691	0.654	0.617	0.630
15	Vote	16	300	0.188	0.250	**0.175**	0.285	0.326	0.292	0.330
16	WaveformEW	40	5000	**0.480**	0.620	0.560	0.847	0.593	0.654	0.567
17	WineEW	13	178	0.446	0.446	0.431	0.526	0.444	0.470	**0.389**
18	Zoo	16	101	0.375	0.475	0.388	0.441	**0.361**	0.458	0.396

Bold values indicate the best results

binary bits of decision variables in the problem. This can improve the exploration of algorithm when the frequency of changing binary bits is high. However, it might interrupt other mechanism of algorithms and degrades the accuracy of the final approximated solution. The results of this section show that a memtic algorithm can be more beneficial since the best feature of algorithms can be combined to better solve a problem.

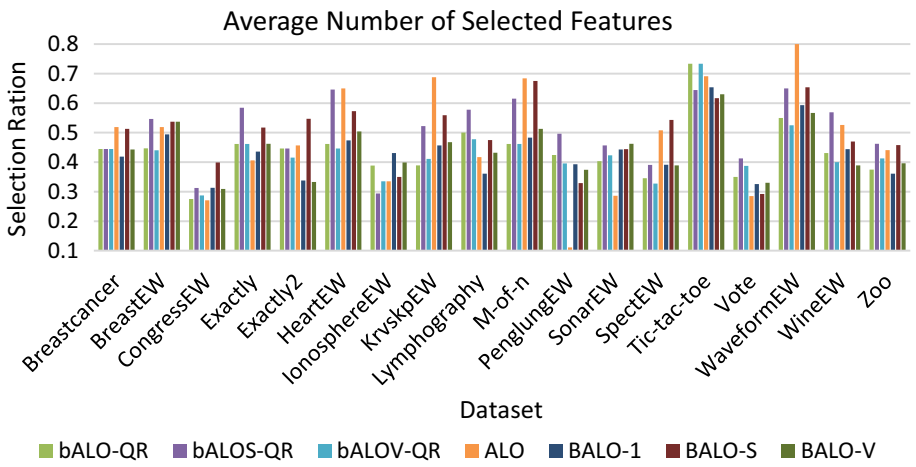


Fig. 2 Average number of selected features by QuickReduct-based approaches and other optimizers

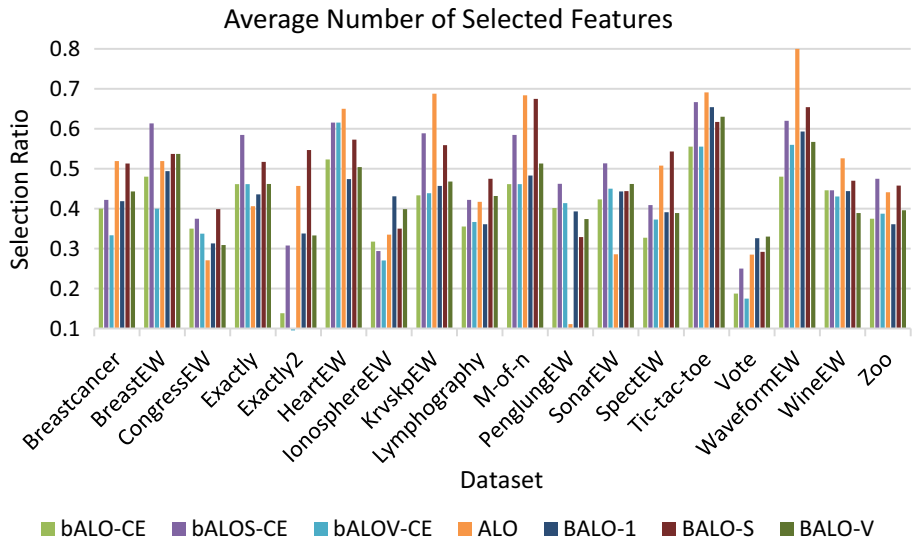


Fig. 3 Average number of selected features by CEBARKCC-based approaches and other optimizers

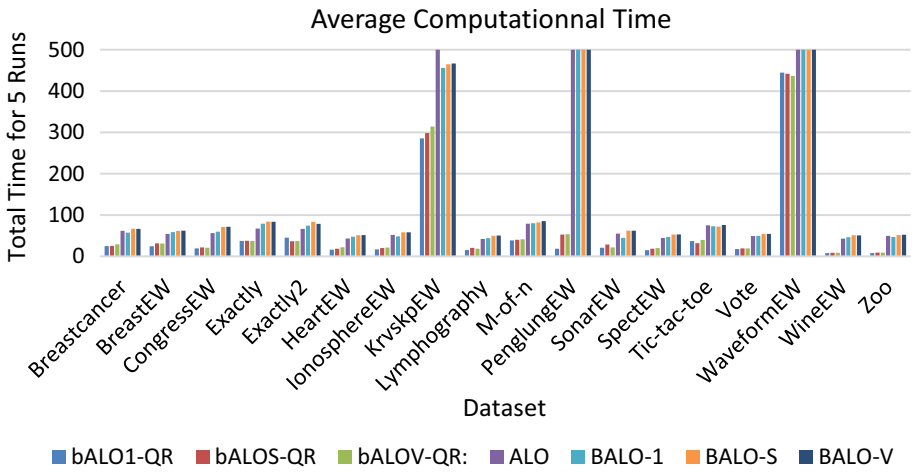


Fig. 4 Total time for QuickReduct approaches and other optimizers

6 Conclusion

This paper presented two variants of a hybrid Ant Lion Optimizer for feature selection called (HBALO) where a hybridization model between ALO and two incremental hill-climbing algorithms namely QuickReduct and CEBARKCC was proposed. QuickReduct algorithm is a rough set-based filter feature selection method that simulates the forward gen-

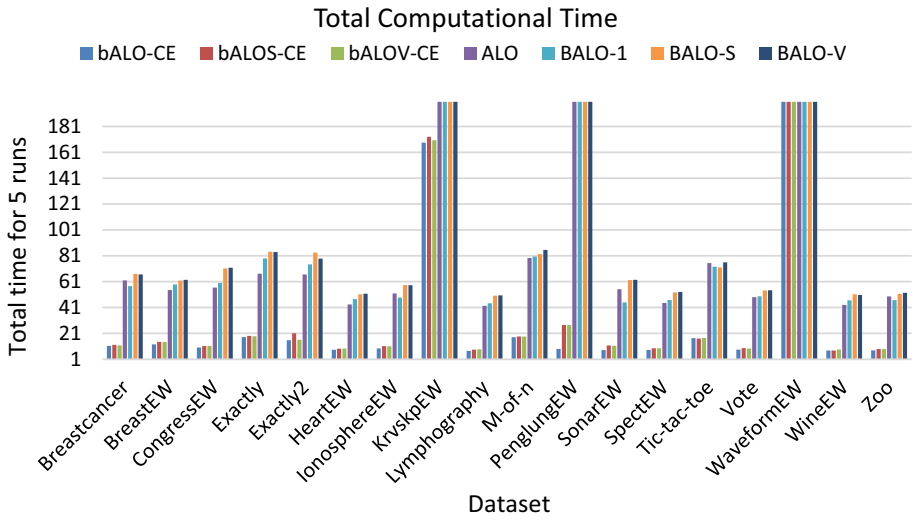


Fig. 5 Total time for CEBARKCC approaches and other optimizers

Table 12 Total computational time for QuickReduct-based approaches and other different optimizers averaged over the different initializers

DS no.	Dataset	bALO1-QR	bALOS-QR	bALOV-QR	ALO	BALO-1	BALO-S	BALO-V
1	Breastcancer	**24.942**	25.406	29.118	62.006	57.561	67.067	66.571
2	BreastEW	**24.509**	31.653	31.016	54.539	58.863	61.860	62.393
3	CongressEW	**19.528**	21.672	21.152	56.461	59.967	71.188	71.825
4	Exactly	**37.324**	37.977	37.641	67.170	78.934	84.121	83.861
5	Exactly2	45.448	**36.708**	37.530	66.609	74.356	83.471	78.834
6	HeartEW	**16.329**	18.757	22.111	43.311	47.528	51.228	51.639
7	IonosphereEW	**16.914**	20.267	21.528	51.923	48.638	58.363	58.286
8	KrvskpEW	**285.617**	298.467	314.330	560.125	456.039	464.636	466.768
9	Lymphography	**15.293**	20.724	18.711	42.352	44.308	50.204	50.405
10	M-of-n	**38.596**	40.047	41.328	79.271	80.473	82.328	85.512
11	PenglungEW	**18.630**	52.698	53.478	8598.746	6435.019	6542.570	6668.146
12	SonarEW	**20.894**	28.935	21.741	55.097	45.025	62.185	62.402
13	SpectEW	**15.246**	18.733	20.021	44.499	46.777	52.721	53.075
14	Tic-tac-toe	36.895	**31.796**	39.662	75.378	72.649	71.961	75.947
15	Vote	**17.902**	19.501	19.210	49.078	49.682	54.245	54.341
16	WaveformEW	444.731	441.899	**436.883**	1685.168	1256.171	1230.965	1276.610
17	WineEW	**7.969**	8.808	8.654	42.959	46.632	51.161	50.704
18	Zoo	**7.875**	9.345	9.302	49.572	46.751	51.571	52.328

Bold values indicate the best results

eration method where the algorithm starts from an empty set and only the features that improve the fitness value will be added. CEBARKCC is a conditional entropy-based method that finds the core features and adds them to the feature subset. These two methods are hybridized with the ALO algorithm in order to improve the quality of the initial population and eventually final optimal solution. The proposed approaches were tested over 18 well-known UCI datasets and compared among well-known feature selection methods: PSO, GA and continuous ALO. Different aspects of performance were assessed to evaluate the proposed approaches. For one, results show that the QuickReduct-based approach performs better than the CEBARKCC approach in terms of classification accuracy while the latter outperforms the former in terms of the minimal reducts. For another, both HBALO methods outperform other approaches in the majority of case studies. This shows the ability of the proposed approach to search the space of features adaptively and its capability to balance between exploration and exploitation efficiently. We can conclude that the quality of the initial population affects the search capability of the optimization algorithm. As the proposed approach shows a good performance when using a good initial population, we recommend further enhancement of the ALO algorithm using other local search algorithms.

Table 13 Total computational time for CEBARKCC-based approaches and other different optimizers averaged over the different initializers

DS No.	Dataset	bALO-CE	bALOS-CE	bALOV-CE	ALO	BALO-1	BALO-S	BALO-V
1	Breastcancer	**11.307**	12.107	11.750	62.006	57.561	67.067	66.571
2	BreastEW	**12.555**	14.499	14.369	54.539	58.863	61.860	62.393
3	CongressEW	**10.095**	11.189	11.364	56.461	59.967	71.188	71.825
4	Exactly	**18.130**	19.108	18.709	67.170	78.934	84.121	83.861
5	Exactly2	**15.768**	21.031	16.076	66.609	74.356	83.471	78.834
6	HeartEW	**8.356**	9.188	9.289	43.311	47.528	51.228	51.639
7	IonosphereEW	**9.374**	11.173	11.032	51.923	48.638	58.363	58.286
8	KrvskpEW	**168.453**	173.027	170.298	560.125	456.039	464.636	466.768
9	Lymphography	**7.474**	8.513	8.719	42.352	44.308	50.204	50.405
10	M-of-n	**18.005**	18.678	18.542	79.271	80.473	82.328	85.512
11	PenglungEW	**8.922**	27.502	27.393	8598.746	6435.019	6542.570	6668.146
12	SonarEW	**8.120**	11.747	11.475	55.097	45.025	62.185	62.402
13	SpectEW	**8.241**	9.487	9.547	44.499	46.777	52.721	53.075
14	Tic-tac-toe	17.303	**17.025**	17.469	75.378	72.649	71.961	75.947
15	Vote	**8.420**	9.580	9.328	49.078	49.682	54.245	54.341
16	WaveformEW	**433.546**	445.087	441.927	1685.168	1256.171	1230.965	1276.610
17	WineEW	**7.766**	**7.766**	8.612	42.959	46.632	51.161	50.704
18	Zoo	**7.929**	8.940	8.946	49.572	46.751	51.571	52.328
	Total	**779.763**	835.647	824.844	12948.000	9819.000	10418.000	10586.000

Bold values indicate the best results

Compliance with ethical standards

Conflict of interest All authors declare that there is no conflict of interest.

Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abualigah LM, Khader AT, Hanandeh ES (2017) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J Comput Sci. https://doi.org/10.1016/j.jocs.2017.07.018

Anusha M, Sathiaseelan J (2015) Feature selection using K-means genetic algorithm for multi-objective optimization. Procedia Comput Sci 57:1074–1080

Asir D, Appavu S, Jebamalar E (2016) Literature review on feature selection methods for high-dimensional data. Int J Comput Appl 136(1):9–17

Bell DA, Wang H (2000) A formalism for relevance and its application in feature subset Selection. Mach Learn 41(2):175–195. https://doi.org/10.1023/A:1007612503587

Bello R, Nowe A, Caballero Y, Gómez Y, Vrancx P (2005) A model based on ant colony system and rough set theory to feature selection. Paper presented at the Proceedings of the 2005 conference on genetic and evolutionary computation, Washington DC, USA

Bello R, Gomez Y, Nowe A, Garcia MM (2007) Two-step particle swarm optimization to solve the feature selection problem. Paper presented at the Proceedings of the seventh international conference on intelligent systems design and applications, Brazil

Blake CL, Merz CJ (1998) UCI repository of machine learning databases. Retrieved 1 June, 2016, from http://www.ics.uci.edu/mlearn/

BoussaïD I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117

Chakraborty B (2008) Feature subset selection by particle swarm optimization with fuzzy fitness function. Paper presented at the 3rd International conference on intelligent system and knowledge engineering, 2008. ISKE 2008

Chen Y, Miao D, Wang R (2010) A rough set approach to feature selection based on ant colony optimization. Pattern Recogn Lett 31(3):226–233

Chen H, Jiang W, Li C, Li R (2013) A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. Math Probl Eng. https://doi.org/10.1155/2013/524017

Christaline JA, Ramesh R, Vaishali D (2016) Bio-inspired computational algorithms for improved image steganalysis. Indian J Sci Technol. https://doi.org/10.17485/ijst/2016/v9i10/88995

Chuang L-Y, Chang H-W, Tu C-J, Yang C-H (2008) Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 32(1):29–38

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B Cybern 26(1):29–41

Emary E, Zawbaa HM (2016) Impact of chaos functions on modern swarm optimizers. PLoS ONE 11(7):e0158738

Emary E, Zawbaa HM (2018) Feature selection via Lèvy Antlion optimization. Pattern Anal Appl. https://doi.org/10.1007/s10044-018-0695-2

Emary E, Zawbaa HM, Hassanien AE (2016a) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65

Emary E, Zawbaa HM, Hassanien AE (2016b) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381. https://doi.org/10.1016/j.neucom.2015.06.083

Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning. Springer series in statistics. Springer, Berlin

Goodenough J, McGuire B, Jakob E (2009) Perspectives on animal behavior. Wiley, Hoboken

Grosan C, Emary E, Zawbaa H (2018) Experienced grey wolf optimizer through reinforcement learning and neural networks. IEEE Trans Neural Netw Learn Syst (TNNLS) 29(13):681–694. https://doi.org/10.1109/TNNLS.2016.2634548

Gunasundari S, Janakiraman S, Meenambal S (2016) Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis. Expert Syst Appl 56:28–47

Holland JH (1992) Adaptation in natural and artificial systems. MIT Press, Cambridge

Hutchins M, Olendorf D (2004) Grzimek's animal life encyclopedia: lower metazoans and lesser deuterosomes, vol 1. Gale/Cengage Learning, Farmington Hills

Il-Seok O, Jin-Seon L, Byung-Ro M (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437. https://doi.org/10.1109/TPAMI.2004.105

Jensen R, Shen Q (2002) Fuzzy-rough sets for descriptive dimensionality reduction. Paper presented at the Proceedings of the 2002 IEEE international conference on fuzzy systems, 2002. FUZZ-IEEE'02

Jensen R, Shen Q (2003) Finding rough set reducts with ant colony optimization. Paper presented at the Proceedings of the 2003 UK workshop on computational intelligence

Jensen R, Shen Q (2004) Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Trans Knowl Data Eng 16(12):1457–1471. https://doi.org/10.1109/TKDE.2004.96

Jensen R, Shen Q (2008) Computational intelligence and feature selection: rough and fuzzy approaches. Wiley-IEEE Press, Hoboken

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University Press, Erciyes. Kayseri/Türkiye: Erciyes University, Engineering Faculty, Computer Engineering Department

Ke L, Feng Z, Ren Z (2008) An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recogn Lett 29(9):1351–1357

Kennedy J, Eberhart R (1995) Particle swarm optimization. Paper presented at the Proceedings of the IEEE international conference on neural networks, 1995

Kittler J (1975) Mathematical methods of feature selection in pattern recognition. Int J Man Mach Stud 7(5):609–638

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1):273–324

Liu H, Motoda H (1998) Feature selection for knowledge discovery and data mining. Kluwer, Boston

Mafarja M, Abdullah S (2013) Investigating memetic algorithm in solving rough set attribute reduction. Int J Comput Appl Technol 48(3):195–202

Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312

Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453

Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Ala'M A-Z, Mirjalili S (2017a) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl Based Syst 145:25–45

Mafarja M, Eleyan D, Abdullah S, Mirjalili S (2017b) S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. Paper presented at the Proceedings of the international conference on future networks and distributed systems

Mafarja M, Jaber I, Eleyan D, Hammouri A, Mirjalili S (2017c) Binary dragonfly algorithm for feature selection

Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Moradi P, Gholampour M (2016) A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl Soft Comput 43:117–130

Osman IH, Kelly JP (2012) Meta-heuristics: theory and applications. Springer, Berlin

Pawlak Z (1982) Rough sets. Int J Inf Comput Sci 11:341–356

Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer, Dordrecht

Shokouhifar M, Sabet S (2010) A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization. Paper presented at the 3rd International conference on machine vision (ICMV 2010)

Ślezak D (2002) Approximate entropy reducts. Fundamenta informaticae 53(3–4):365–390

Theodoridis S, Koutroumbas K (2006) Pattern recognition, 3rd edn. Academic Press, Orlando

Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 28(4):459–471

Wang J, Li T, Ren R (2010) A real time IDSs based on artificial bee colony-support vector machine algorithm. Paper presented at the 2010 Third international workshop on advanced computational intelligence (IWACI)

Wang H, Khoshgoftaar TM, Napolitano A (2012) Software measurement data reduction using ensemble techniques. Neurocomputing 92:124–132

Wang A, An N, Chen G, Li L, Alterovitz G (2015a) Accelerating wrapper-based feature selection with K-nearest-neighbor. Knowl-Based Syst 83:81–91

Wang Y, Liu Y, Feng L, Zhu X (2015b) Novel feature selection method based on harmony search for email classification. Knowl-Based Syst 73:311–323

Wolpert D (1997) No free lunch theorem for optimization. IEEE Trans Evol Comput 1:467–482

Xue B, Zhang M, Browne WN (2014) Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Appl Soft Comput 18:261–276

Yang J, Honavar VG (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 13(2):44–49. https://doi.org/10.1109/5254.671091

Yu H, Wang G, Yang D, Wu Z (2002) Knowledge reduction algorithms based on rough set and conditional information entropy. Paper presented at the AeroSense 2002

Zawbaa HM, Emary E, Parv B (2015) Feature selection based on antlion optimization algorithm. Paper presented at the 2015 Third world conference on complex systems (WCCS)

Zawbaa HM, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. PLoS ONE 11(3):e0150652

Zawbaa HM, Emary E, Grosan C, Snasel V (2018) Large-dimensionality small-instance set feature selection: a hybrid bio-inspired heuristic approach. Swarm Evolut Comput. https://doi.org/10.1016/j.swevo.2018.02.021