Expert Systems with Applications 159 (2020) 113572

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Enhanced Crow Search Algorithm for Feature Selection

Salima Ouadfel [a], Mohamed Abd Elaziz [b,c,*]

[a] Department of Computer Science, NTIC Faculty, University of Constantine2, Abdelhamid Mehri, 25000 Constantine, Algeria
[b] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, China

ARTICLE INFO

Article history:
Received 28 December 2019
Revised 26 March 2020
Accepted 14 May 2020
Available online 6 June 2020

Keywords:
Feature selection
Crow search algorithm (CSA)
Metaheuristic
Dynamic local neighborhood
Adaptive awareness probability

ABSTRACT

The crow search algorithm (CSA) is a recent metaheuristic inspired by the intelligent group behavior of crows. It has attracted the attention of many researchers because of its simplicity and easy implementation. However, it suffers from premature convergence because of its ability to balance between exploration and exploitation is weak. Therefore, we investigate in this paper, an enhanced version of CSA called by us ECSA as a wrapper feature selection method to extract the best feature subsets. This enhancement achieved by introducing three modifications to the original CSA to improve its performance. Firstly, we propose an adaptive awareness probability to enhance the balance between exploration and exploitation. Secondly, we replace the random choice of the crow to follow by the dynamic local neighborhood to guide the local search. Thirdly, we introduce a novel global search strategy to increase the global exploration capability of the crow. The performance of ECSA is measured using three performance metrics and statistical significance over 16 datasets from the UCI repository. The obtained results are compared with those of the original CSA and some state-of-the-art techniques in the literature. Experimental results showed that ECSA presents a better convergence speed and a better-quality solution.

1. Introduction

Feature selection (FS) aims to select the optimal subset of relevant features from a large number of features (Ekbal & Saha, 2015; Guyon, Gunn, Nikravesh, & Zadeh, 2006) by selecting the smallest number of relevant and non-redundant ones (Xue, Zhang, Browne, & Yao, 2016). According to these properties of FS, it has been applied to different applications including text intrusion detection (Lee, Stolfo, & Mok, 2000), customer relationship management (Ng & Liu, 2000), genomic analysis (Xing, Jordan, & Karp, 2001) and image retrieval (Rui, Huang, & Chang, 1999; Swets & Weng, 1995).

In the literature, there are three different approaches of feature selection: wrapper-based methods, filter-based methods and embedded methods (Dash & Liu, 1997; Guyon et al., 2006; Xue et al., 2016). The wrapper-based approach selects the best subset of features using a classifier. On the other hand, a filter-based approach searches the optimal subset features according to some criteria. It takes into account only the data without any use of a classifier. In the third approach, namely, the embedded approach, the search occurs through the integrating feature selection and classifier learning into a single process (Xue et al., 2016).

Feature selection problem can be considered as an optimization problem when the feature space is too large. In fact, in the case of an original dataset with N features, there are 2^{N} possible permutations to select the optimal subset of features. When the original number of features N is too large, the search space grows exponentially, and it becomes impractical to use an exhaustive search for feature selection (Dash & Liu, 1997). To tackle this problem, metaheuristics have been introduced to limit the exploration of the features space (Rodrigues et al., 2015). In this field, we find particle swarm optimization (PSO) (Chuang, Yang, Wu, & Yang, 2011; Xue, Zhang, & Browne, 2014), Artificial bee colony (ABC) (Hancer, Xue, Karaboga, & Zhang, 2015), Differential Evolution (DE) (Chattopadhyay, Mishra, & Goswami, 2016; Maheshwari, Kumar, & Kumar, 2016; Sikdar, Ekbal, Saha, Uryupina, & Poesio, 2015), Bat algorithm (BA) (Nakamura et al., 2012), Month flame Optimization (MFO) (Zawbaa, Emary, Parv, & Sharawi, 2016), Grey Wolf Optimizer (GWO) (Emay, Zawbaa, & Hassnien, 2016), bufferfly optimization algorithm (BOA) (Arora & Anand, 2019), Sine Cosine Algorithm (SCA) (Abd Elaziz, Oliva, & Xiong, 2017; Sindhu, Ngadiran, Yacob, Zahri, & Hariharan, 2017) and Crow Search Algorithm (CSA) (De Souza, dos Santos Coelho, De Macedo, & Pierezan, 2018; Sayed, Hassanien, & Azar, 2017).

Crow search Algorithm (CSA) is one of the newly proposed algorithms that have attracted considerable attention in the past years

* Corresponding author.
E-mail address: abd_el_aziz_m@yahoo.com (M. Abd Elaziz).

https://doi.org/10.1016/j.eswa.2020.113572
0957-4174/

(Askarzadeh, 2016). The main concept behind the CSA is that crows store their food in hiding places to retrieve it when they need and will follow other crows to cheat their stored food. In (Askarzadeh, 2016), CSA has been applied to optimize six constrained engineering design problems, and the results obtained reveal that using CSA may lead to finding promising results compared to the other optimization algorithms. Due to its simplicity, easy implementation, efficiency, and has few parameters, so it has been used by many researchers since its development (Dos Santos Coelho, Richter, Mariani, & Askarzadeh, 2016; Kumar, Pakhali, Trivedi, & Jangir, 2016; Shaheen & El-Sehiemy, 2017). However, like any metaheuristic, CSA can be trapped in a local optimum because of a lack of the exploitation and exploration processes (Shi, Li, Zhang, & Huang, 2017). To deal with this drawback, some modifications have been introduced to the original CSA. For example, in (Shi et al., 2017), authors use an adaptive inertia weight factor and roulette wheel selection scheme to enhance the exploration and exploitation capabilities of CSA. In (Díaz et al., 2018), the fixed awareness probability (AP) value is adjusted dynamically according to the fitness value of each candidate solution, and the Lévy flight movement is also used to enhance the search capacities of the CSA. In (Sayed et al., 2017) CSA was embedded with ten chaotic maps to improve its performance. In (Javaid et al., 2017; Javaid, Mohsin, Iqbal, Yasmeen, & Ali, 2018), CSA is hybridized with BA and DE to enhance the performance of the original CSA. In (Majhi, Sahoo, & Pradhan, 2019), authors use the Opposition based learning (OBL) instead of the random initialization for improving the convergence speed and a random normal mutation operator to increase the performance of CSA algorithm. In (Qu & Fu, 2019), a new crow search algorithm based on neighborhood search of non-inferior solution set (NICSA) is proposed. In (Zamani, Nadimi-Shahraki, & Gandomi, 2019), authors introduce three search approach to enhance the performance of CSA. These methods are the non-neighborhood based global search (NGS), neighborhood-based local search (NLS), and wandering around based search (WAS). However, these methods have some limitations that effect on the quality of the final solution. According to No-Free-Lunch (NFL) theorem (Wolpert & Macready, 1997), there is no algorithm better than all other algorithms with all classes of feature selection problems. Hence, there is a need to propose new algorithms or to improve the original ones by the introduction of some modification to their operators to tackle feature selection problems more efficiently. So, this motivated us to propose an enhanced version of CSA, named ECSA, which aims to increase the performance of the original algorithm. We embedded ECSA with an adaptive AP to obtain a better compromise between exploitation and exploration. A dynamic local neighborhood is used to guide the local search, and a modified global search strategy is proposed to increase the exploration ability of CSA. In ECSA, instead of a unique fixed value for AP, a dynamic updating strategy is applied for choosing the values of AP of each row based on its ranking in the population. According to the value of AP, a local or a global search is performed by each crow. In the local search strategy, a dynamic local neighborhood formed by the local neighbors of a crow is constructed, and the crow to follow is chosen randomly from it. In the global search strategy, a random search is performed around the best global position. Also, to the best of our knowledge, there are only two papers about the application of CSA for feature selection. In (Sayed et al., 2017), CSA was embedded with ten chaotic maps to enhance its performance, and in (De Souza et al., 2018) a transfer function is used to transform the continuous positions of crows to binaries ones without any modification to the CSA operators.

The proposed ECSA begins by setting the initial positions for a set of *NP* crows then each solution is converted to a binary solution that represents the set of selected features. The next step is evaluating the performance of this subset using a fitness function (we used the accuracy of classification). Thereafter, the best solution is determined, and the solutions will be updated using the modified CSA. The process of updating is performed using until it reached the stop conditions. The best solution is the returned output.

The main contributions of the current study can be formulated as:

1. Provide an alternative feature selection method based on a modified crow search algorithm using an adaptive AP, a dynamic local neighborhood strategy and a new global search mechanism.
2. Adjust the value of AP for a crow during the search process according to its position in the individual ranking to balance the exploitation of good solutions and exploration of new ones.
3. Introduce a local neighborhood inspired from the PSO to guide the selection of the crow to follow and therefore improve the exploitation of good local solutions
4. Improve the global search by the introduction of new global operators, which guide the search direction of crows around the best solution found so far.
5. Evaluate the performance of the proposed ECSA using a set of UCI machine datasets and Compare the proposed ECSA with other state-of-the-art FS approaches.

The paper is organized as follows: In Section 2, related works are presented. In Section 3, we present the classical CSA, and in Section 4, the proposed Enhanced CSA (ECSA) is described. Experimental results and discussions are introduced in Section 5. In Section 6, the conclusion and future works are given.

2. Related works

During the last decades, many metaheuristics have been extensively used as search strategies in feature selection methods and demonstrate superior efficiencies when compared to the exact methods (Zorarpacı & Özel, 2016). For instance, in (Kashef & Nezamabadi-pour, 2015), authors propose an ACO-based FS algorithm where all features are represented as nodes of a graph and ants move from a node to another to select the best features subset. Artificial bee colony (ABC) algorithm has been hybridized with ACO and DE respectively to take advantage of each one to find the optimal subset of features (Shunmugapriya & Kanmani, 2017; Zorarpacı & Özel, 2016). A binary version of the bat algorithm (BA) is introduced as a wrapper approach for solving the feature selection problem (Nakamura et al., 2012). In (Djellali, Djebbar, Zine, & Azizi, 2018) two hybrid versions of the ABC algorithm with the PSO and GA algorithms are introduced such that during the search procedure, Particle swarm PSO contributes to ABC during the employed phase, and GA mutation operators are applied in the Onlooker phase and Scout phase. Another hybridization of ABC was proposed in (Rao et al., 2019), ABC algorithm with the gradient boosting decision tree. The PSO algorithm has been widely used for feature selection problem. In (Moradi & Gholampour, 2016), a hybrid version of the PSO with a local search is proposed to select the best feature subset. In (Mafarja & Sabar, 2018), authors propose a new approach based PSO for feature selection using an adaptive update strategy based on the rank of each particle in the swarm to adapt the value of the inertia weight parameter dynamically. In (Pourpanah, Shi, Lim, Hao, & Tan, 2019), authors combine the Fuzzy ARTMAP (FAM) model with BSO for solving the feature selection problem. In (Chen et al., 2017), the authors propose a novel approach based on bacterial foraging optimization (BFO) for feature selection. An enhanced version of the cuckoo search combined with rough sets was proposed in (El Aziz & Hassanien, 2016). In (Maheshwari et al., 2016) DE algorithm and GA were used for feature selection tasks to enhance face recognition accuracy. Authors in (Sharawi, Zawbaa, & Emary, 2017), a binary whale optimization algorithm (WOA) has been employed as a search strategy in a wrapper FS method. In (Dash, 2018), the harmony search algorithm (HS) is embedded with the Pareto optimization for feature selection in high dimensional data classification problems. In (Too, Abdullah, Mohd Saad, Mohd Ali, & Tee, 2018), a competitive binary grey wolf algorithm (GWO) is suggested for the optimal feature subset from the original feature set for the classification purposes of electromyography signals. A binary Sine Cosine Algorithm (SCA) based feature selection approach was proposed by (Sindhu et al., 2017). The results were better than those of PSO and GA, among ten datasets. In (Faris et al., 2018), the Salp Swarm Algorithm (SSA) was embedded with a new crossover operator and applied for the feature selection problem. The results were better than the basic version of SSA. In (Sayed et al., 2017), ten chaotic maps are embedded with the CSA to select the optimal features from the full features set of 20 benchmark datasets. In (De Souza et al., 2018), a new wrapper based on v-shaped binarization of the CSA is proposed to solve the feature selection problem. In (Mafarja et al., 2019), authors develop three hybrid models of the GWO and WOA to take advantage of each optimizer. In the hybrid serial GWO-WOA (HSGW), each solution is firstly modified using the GWO operators and secondly refined using the WOA ones. In the random switcher GWO-WOA (RSWGO), in each solution, WOA or WOA is selected randomly to update the population. In adaptive switcher WOA-WOA (ASGO), an adaptive mechanism is introduced for the choice of the operators. The selection is based on the number of solutions improved by each optimizer. From the experimental results, it has been observed that the HSWO is the best performing optimizer. In (Mafarja et al., 2018), a GOA-based optimizer with EPD (Evolutionary Population Dynamic) and the new selection and mutation operators are proposed to improve the performance of the classical GOA algorithm.

3. Overview of crow search algorithm (CSA)

The Crow Search Algorithm (CSA) is a new population-based metaheuristic introduced by (Askarzadeh, 2016). The position of each individual crow i at generation gen in the search space can be specified by a vector x_i^{gen} and represents a possible solution to the problem under study.

$$x_i^{gen} = \left[x_{i,1}^{gen}, x_{i,2}^{gen}, \ldots, x_{i,n}^{gen}\right]; \qquad i = 1, 2, \ldots., NP; \quad gen = 1, \ldots., \mathrm{max\_gen} \tag{1}$$

where NP is the population size, and max_gen is the maximum number of generations.

Each crow has a memory m_i^{gen} that represents the position of its hiding food, and at the same time, it is considered as the best position of that crow (Askarzadeh, 2016). During each generation, crows move in the search space and try to steal the hidden food of others (Askarzadeh, 2016). At each generation gen, crow i moved to a new position according to the following equation (Askarzadeh, 2016):

$$x_i^{gen+1} = \begin{cases} x_i^{gen} + r_i \times FL_i^{gen}\left(m_j^{gen} - x_i^{gen}\right) & \text{if } a_j \geqslant AP_j^{gen} \\ \text{a random position} & \text{otherwise} \end{cases} \tag{2}$$

where r_i and a_j are a random number with uniform distribution in [0. 1] and AP_j^{gen} is the Awareness Probability of crow j at generation gen (Askarzadeh, 2016), FL_i^{gen} is the flight length of crow i at generation gen, and m_j^{gen} denotes the memory of crow j at generation gen. FL_i^{gen} Significantly affects the searching capability of the algorithm. Lower values of FL promote local search, and higher ones lead to global search (Askarzadeh, 2016; Hassanien, Rizk-Allah, & Elhoseny, 2018).

Initially, each crow's position x_i^{gen}(gen = 0) is placed randomly at the search space. In this step, we consider that initial positions also represent the memories of the crows. After the initial step, the CSA relocates each crow individual, using Eq. (2).

During each generation, the new position' crow is evaluated using an objective function *fit*, and then, the crows update their memorized positions according to Eq. (3) (Askarzadeh, 2016).

$$m_i^{gen+1} = \begin{cases} x_i^{gen+1} & \text{if } fit(x_i^{gen}) \text{ is better than } fit(m_i^{gen}) \\ m_i^{gen} & \text{otherwise} \end{cases} \tag{3}$$

The pseudo-code of the CSA is presented below (Askarzadeh, 2016):

Algorithm 1. Crow Search Algorithm (CSA)

```
Set parameters of CSA:
    max_gen: maximum number of generations;
    NP: size of the flock of crows;
    FL: flight light of crows;
    AP: Awareness Probability
Set gen = 0;
Initialize randomly positions of NP crows
Initialize the memory of each crow
Evaluate the position of each crow
while gen < gen_max
  for i = 1 : NP (all NP crows of the flock)
Randomly select one crow to follow (for example j)
    Update crow's position using Eq. (2)
  end for
  Check new positions according to the constrains of the
   problem
  Assess the updated position of the crows
  Using Eq. (3) to update the memory of crows
  gen = gen + 1
end while
Return the best solution
```

4. Proposed enhanced CSA (ECSA)

Like any metaheuristic, CSA suffers from a good balance between exploitation and exploration, so it can be trapped in a local optimum (Shi et al., 2017). Its performances in terms of convergence and search capability are mainly related to the AP parameter and the choice of the crow to follow. In this section, we describe our proposed enhanced CSA (ECSA) to tackle this weakness. In order to enhance the global searching and local searching abilities, we propose a new improved CS in three aspects: a dynamic awareness probability strategy (DAP) to balance diversification and intensification, a local neighborhood searching strategy (LNSS) to select the crow to follow and novel global updating position strategy.

4.1. Dynamic awareness probability (DAP)

The main advantage of CSA is that it has few parameters to adjust: namely flight length FL and awareness probability AP. The performance of CSA is greatly affected by the value of AP since it controls the balance between exploration and exploitation. In the classical CSA, the value of AP is constant and fixed to 0.1 for all crow since the beginning of the optimization process, which can lead to poor results.

To improve the selection of the exploration or exploitation strategy in the classical CSA, we propose in this paper, to modify the value of AP of each crow for each iteration depending on its rank by using a dynamic updating strategy for AP (DAP) inspired from the work of (Gong & Cai, 2013). In this strategy, the fitness function is evaluated for each crow. Then, a sorting operation is performed in ascending order from the best to the worst based on the fitness value of each crow. After this sorting, each crow is assigned a rank using Eq. (4):

$$rank_{\,i} = i, \;\; i = 1, 2, \ldots., NP \tag{4}$$

According to Eq. (4), the crow with the lower fitness (the better solution) will obtain rank one, and the worst solution will be given the last rank.

The AP assigned to each individual in the swarm depends only on its position in the individual rank and not on its fitness value. For each crow the AP value is selected between AP_{min} and AP_{max} according to its rank as follows:

$$DAP_{\mathrm{i}}^{gen} = AP_{\min} + (AP_{\max} - AP_{\min})\frac{rank_i}{NP} \tag{5}$$

where $rank_i$ is the rank of the ith solution, NP refers to the size of the population. $AP_{\min}$ and $AP_{\max}$ are the minimum and maximum values of AP. In this work, we set experimentally $AP_{\min}$ to 0.1 and $AP_{\max}$ to 0.8.

According to the above equation, we can see that the AP value increases with the increase in the rank of the crow. In this manner, the best crow takes the first rank, and so its AP is set to the minimum value of AP while the worst crow takes the last rank, and its AP is set to a maximum value of AP. The best crow has a high probability of performing a local search (exploitation) while the worst crow performs a global search (exploration by random movement around the global best solution).

For each crow X_i and according to the value of its DAP, a local or a global search strategy will be adopted as follows:

if $rand \geqslant DAP_{\mathrm{j}}^{gen}$
Do Local search
else
Do Global search
end

4.2. Improved local search

Let us consider $p = [X_1, \ldots X_{NP}]$. Each crow X is a D-dimensional parameter vector $X_i = [x_{1,1}, \ldots x_{i,D}]$. In the original CSA, the position updating of a crow is based on a randomly selected crow's food position. This strategy can cause slow convergence in the case where the selected crow is not a good solution.

To improve the exploitation strategy of the classical CSA, each a crow j chooses a crow i to follow by using a local neighborhood selection strategy (LNSS). To update its position, a crow X_i uses the crow to follow from a small neighborhood rather than the entire population like in the classical CSA. Inspired by the neighborhood's models proposed for PSO, we assume the vectors to be organized on a ring topology with respect to their indices. For each crow i we define a local neighborhood of radius k such that: $X_{i-k}, \ldots X_i, \ldots, X_{i+k}$ are the immediate neighbors of crow X_i. Following the suggestion of (Wang et al., 2011), we set the radius of the neighborhood k = 2 in this paper. Fig. 1 presents the concept of a local neighborhood for the crow X_4.

In ECSA, each crow X_i created from its local neighborhood, a vector a D-dimensional parameter vector $d = [d_1, \ldots d_D]$ such that

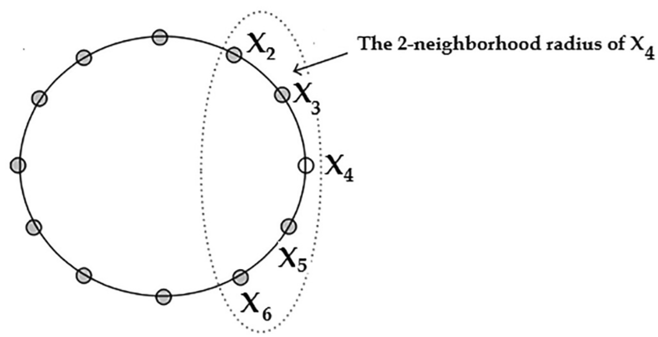

Fig. 1. Ring topology neighborhood in ECSA.

each d_i is the indices of which crow X_j to learn from its best position. After generating the neighborhood of crow X_i, a crow X_j among its local neighbor is randomly selected for each dimension. This means that we are not updating all the dimensions of a crow X_i from the same crow X_j which assures a greater exploration and makes ECSA less prone to being trapped in a local minimum. The crow positions are updated as shown in Eq. (6).

$$x_{i,s}^{gen+1} = x_{i,s}^{gen} + FL^{gen} * (m_{d(j),s}^{gen} - x_{i,s}^{gen}) \;;\, \mathrm{s} = 1 \ldots \mathrm{D} \tag{6}$$

Note that the neighborhood is no static during the searching process. It is changed after a fixed number of generations by doing a random rearrange of the crows in the population so that the exploration ability of the crows is improved by sharing new information.

4.3. Improved global search strategy

In the original CSA, when the ith crow knows about the presence of jth crow, he intentionally takes a random trajectory that can affect the convergence speed. To avoid this issue, we introduce in ECSA a new global strategy to improve the exploration performance of CSA inspired by (Aljarah et al., 2018). According to Eq. (7), we update the position of the crow with respect to the global best solution obtained so far, so the crow explores and exploits the space around it.

$$\begin{aligned} & if\ rand < 0.5 \\ x_{\mathrm{i}}^{gen+1} &= Best + c1 * c2 \\ & else \\ x_{\mathrm{i}}^{gen+1} &= Best - c1 * c2 \\ & end \end{aligned} \tag{7}$$

where x_i is the current position of the ith crow, and Best is the global best position. c1 is a variable that is linearly decreased over the iterations, and calculated as given in Eq. (8) (Aljarah et al., 2018), c2 is a variable drawn from the interval [0. 1]. Therefore, Eq. (7) allows a crow to update its position in the neighborhood of the global best solution:

$$c_1 = 2 * \exp(-4 * gen/\max\_gen)^2 \tag{8}$$

4.4. ECSA for feature selection problem

The possible presence of the redundant and/or irrelevant features alters the classification accuracy and yields to a slow down classification process. To reduce this issue, feature selection methods are used to find from a set of M features, a subset with size m where $m < M$ without altering the classification accuracy. When M is too large, it is not possible to use an exhaustive approach for examining all possible features. We propose in this paper, an enhanced version of CSA (ECSA) for solving the feature selection problem.

ECSA starts by set the initial value for a set of solutions, where the whole population p^{gen} (gen = 0.. . .,max_gen) of *NP* crows with *M* dimensional is represented by a matrix as shown below:

$$p^{gen} = \begin{bmatrix} x^{gen}_{1,1}\ x^{gen}_{1,2}\ldots x^{gen}_{1,M} \\ \cdot\ \cdot\ \ldots\ \cdot \\ \cdot\ \cdot\ \ldots\ \cdot \\ \cdot\ \cdot\ \ldots\ \cdot \\ x^{gen}_{NP,1}\ x^{gen}_{NP,2}\ldots x^{gen}_{NP,M} \end{bmatrix}$$

Here, matrix p^{gen} stores the positions of crows $x^{gen}_{i,d}$ denotes the location of the *ith* agent in the *dth* dimension at generation *gen*, *NP* denotes the number of crows, and *M* is the number of total features in the dataset.

In the ECSA algorithm, each crow is initially placed at a position chosen randomly as follows:

$$x_{ij} = l_{ij} + rand * (u_{ij} - l_{ij}),\ i = 1, 2, \ldots, NP,\ j = 1, \ldots, M,$$

where l_{ij}, u_{ij} represent the lower boundary and the upper boundary of the element $x_i \in X$ at the *jth* dimension, respectively. Then each crow is converted to binary solutions using the following equation:

$$x^{gen+1}_{i,d} = \begin{cases} 1 & if\ S(x^{gen+1}_{i,d}) \geqslant \sigma \\ 0 & otherwise \end{cases} \tag{9}$$

where $x^{gen+1}_{i,d}$ is the continuous-valued input vector x_i at dimension *d* and generation gen. $\sigma \in [0.1]$ is a random value which represents the threshold. The sigmoid function which is given in Eq. (10)

$$S(x^{gen+1}_{i,d}) = \frac{1}{1 + e^{-x^{gn+1}_{i,d}}} \tag{10}$$

Then the features subset of each crow is evaluated by the KNN classifier. As we aim to find a subset with a minimum number of features and maximum classification accuracy, we use the fitness represented by the following equation:

$$fit = \alpha.Err(S) + (1 - \alpha).\frac{|R|}{|C|} \tag{11}$$

where *Err* (*S*) is the classification error. |R| is the size of the subset features S and |C| is the size of the original features set in the dataset. α is a real parameter $\in$ [1,0] corresponding respectively to the weight of the classification error rate and $(1 - \alpha)$ represents the importance of the feature reduction. In this paper, we set α to 0.9.

Table 1
List of used datasets.

	Dataset	No of features	No of instances
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.	Lymphography	18	148
9.	M-of-n	13	1000
10.	PenglungEW	325	73
11.	SonarEW	60	208
12.	SpectEW	22	267
13.	Tic-tac-toe	9	958
14.	Vote	16	300
15.	WineEW	13	178
16.	Zoo	16	101

Thereafter, the best solution is determined, and the crow positions are updated using local search (as in Eq. (6) or global search (as in Eq. (7) according to the value of DAP, which is calculated using Eq. (5). The whole procedure is repeated until the maximum number of generations is met. Algorithm 2 presents the pseudo-code of ECSA for feature selection.

Algorithm 2. The pseudo-code of the ECSA Algorithm

Load dataset and divide samples into training and test sets.
Set parameters:
    max_gen: maximum number of generations;
    NP: size of the flock of crows;
    M: number of features;
    FL: flight light of crows;
Set *gen* = 1;
Initialize the position of *NP* crows in the search space using Algorithm 2
Initialize the memory of each crow with their initial positions
Evaluate the fitness of each crow using Eq. (9)
Sort population *pop* based on the fitness value;
while *gen* < *max_gen*
  Set c1 using Eq. (8)
  for i = 1 : size(pop)
    Calculate the DAP^{gen}_j using Eq. (5)
    If rand>=DAP^{gen}_j
      //Do local search
      Construct from the local neighborhood of *Xi* the vector *d*
      For j = 1: dimension
        $x^{gen+1}_{i,j} = x^{gen}_{i,j} + FL^{gen} * (m^{gen}_{d(j),j} - x^{gen}_{i,j})$ j = 1...M
      end
    Else
    // Do global search
      if rand < 0.5
        $x^{gen+1}_i = Best + c1 * c2$
      else
        $x^{gen+1}_i = Best - c1 * c2$
      end
    end
  Endfor
  Binarize solutions using Eq. (10)
  Assess the updated positions using Eq. (9)
  If fitness improved then enhance the memory of crows.
  Change the local neighborhood
gen = *gen* + 1
End while
Select the global best solution (the best features subset)

5. Experimental results

To study the performance of ECSA, we have followed the same experimental protocol, followed by many researchers in this field. The experiments are performed through two stages: in the first stage, the performance of ECSA are evaluated against CSA and three improved versions of original CSA: ICSA1 (Shi et al., 2017), ICSA2 (Díaz et al., 2018) and CCSA (Zamani et al., 2019). In the second stage, ECSA is compared with some recent state-of-art metaheuristic methods.

5.1. Datasets description

In this study, sixteen well-known benchmark datasets taken from the UCI data repository (Bache & Lichman, 2013) were used to validate the quality of ECSA. These datasets are used by many researchers for performance comparison in the field of feature selection. The characteristics of these datasets are resumed in Table 1. For evaluation purposes, each dataset is normalized using the max–min normalization method, and we use 5-fold cross-validation for each dataset to generate training and test sets (Faris et al., 2018). Each subset of features obtained by the crow is evaluated suing the KNN classifier (in this paper, we set K = 5).

5.2. Performance measures and common parameter setting

The comparative methods used in this paper are evaluated and compared by using the following popular measures used in many papers for the feature selection problem.

Mean Accuracy: It is the mean of the classification accuracy values for the method over running it$M$ times and is given by Eq. (12):

$$\text{AvgAcc} = \frac{1}{\text{M}}\sum_{i=1}^{\text{M}} Acc^{i} \tag{12}$$

where Acc^{i} is the classification accuracy of the best solution obtained at the run i.

Mean fitness: It is the average of the fitness value of the method over Mruns and it is calculated by Eq. (13)

$$\text{Avg Fit} = \frac{1}{\text{M}}\sum_{i=1}^{\text{M}} fit^{i}_{Best} \tag{13}$$

Mean Feature selection ratio: It is the average of the ratio of the number of selected features to the number of original features over M runs and is given by Eq. (14)

$$AvgFR = \frac{1}{M}\sum_{i=1}^{M} \frac{FSN^{i}}{D} \tag{14}$$

where FSN FSN^{i} is the size of the feature selected subset obtained for the best solution at the run i and D is the dimension (total number of features) of the dataset.

All algorithms used this work are uniformly randomly initialized. In all experiments, the maximum iterations are set to 100, and the population size is set to 10. Additionally, all statistical results are recorded over 20 independent runs. The dimension of cases is equal to the number of features in each experimented dataset. The α parameter, which determines the weight of the classification accuracy in the fitness equation, is set to 0.9. These values are used in the majority of the papers in the literature that deals with the feature selection problem using metaheuristics. The proposed optimizers used in this paper are implemented using the Matlab, and all the experiments were done on a PC with Intel Core ™ i5- 3380 2.90 GHz CPU and 4.0 GB RAM.

5.3. Experimental series 1: Comparison with other CSA methods

In the first stage and in order to study the impact of the investigated improvements of the performance of ECSA for feature selection problem, we compared its performance with that of the original CSA (Askarzadeh, 2016), ICSA1 (Shi et al., 2017), ICSA2 (Díaz et al., 2018) and CCSA (Zamani et al., 2019). The CCSA, CSA, ICSA1, and ICSA2 parameter values are fixed according to their corresponding papers. Statistical results are tabulated in Tables 2–6. For each dataset, the optimal values are made in boldface. For each

Table 2
Results of ***AvgAcc*** and its STD for ECSA and four CSA based approaches for 20 runs.

Datasets	Stat. measure	ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	Mean	**0.9716**	0.9687	0.9702	0.9699	0.9700
	Std	0.0019	0.0019	0.0019	0.0021	0.0020
BreastEW	Mean	**0.9576**	0.9522	0.9512	0.9540	0.9504
	Std	0.0035	0.0026	0.0030	0.0029	0.0041
CongressEW	Mean	**0.9663**	0.9631	0.9556	0.9620	0.9579
	Std	0.0043	0.0035	0.0033	0.0042	0.0029
Exactly	Mean	**1.0000**	0.9985	0.9991	0.8848	0.9992
	Std	0.0000	0.0046	0.0027	0.1055	0.0029
Exactly2	Mean	0.7669	0.7592	**0.7716**	0.7509	0.7708
	Std	0.0129	0.0043	0.0082	0.0105	0.0144
HeartEW	Mean	**0.8296**	0.8169	0.8215	0.8222	0.8167
	Std	0.0101	0.0084	0.0094	0.0106	0.0061
IonosphereEW	Mean	**0.9312**	0.9217	0.9095	0.9197	0.9170
	Std	0.0069	0.0040	0.0051	0.0066	0.0054
Lymphographyt	Mean	**0.8652**	0.8453	0.8510	0.8470	0.8547
	Std	0.0127	0.0090	0.0092	0.0110	0.0071
M-of-n	Mean	**1.0000**	0.9997	0.9990	0.9410	0.9995
	Std	0.0000	0.0009	0.0017	0.0587	0.0017
PenglungEW	Mean	**0.9205**	0.9137	0.9089	0.9130	0.9185
	Std	0.0072	0.0064	0.0067	0.0067	0.0054
SonarEW	Mean	**0.9260**	0.9031	0.9084	0.9053	0.9118
	Std	0.0131	0.0063	0.0061	0.0113	0.0052
SpectEW	Mean	**0.8466**	0.8361	0.8320	0.8373	0.8272
	Std	0.0111	0.0088	0.0056	0.0094	0.0044
Tic-tac-toe	Mean	**0.8424**	0.8269	0.8356	0.8122	0.8146
	Std	0.0045	0.0085	0.0057	0.0166	0.0021
Vote	Mean	**0.9603**	0.9540	0.9458	0.9558	0.9538
	Std	0.0054	0.0028	0.0052	0.0044	0.0045
WineEW	Mean	**0.9846**	0.9801	0.9806	0.9803	0.9823
	Std	0.0040	0.0039	0.0039	0.0050	0.0049
Zoo	Mean	0.9827	0.9832	0.9901	0.9837	**0.9931**
	Std	0.0044	0.0047	0.0045	0.0048	0.0047
AvgAcc		**0.9220**	0.9139	0.9144	0.9024	0.9024

The significance of bold means the best values.

Table 3
Result of ***AvgFit*** and its STD for ECSA and four CSA based approaches.

Datasets	Stat. measure	ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	Mean	**0.0337**	0.0367	0.0370	0.0359	0.0359
	Std	0.0019	0.0014	0.0016	0.0017	0.0023
BreastEW	Mean	**0.0467**	0.0519	0.0558	0.0504	0.0551
	Std	0.0038	0.0025	0.0028	0.0030	0.0042
CongressEW	Mean	**0.0363**	0.0393	0.0496	0.0412	0.0465
	Std	0.0045	0.0031	0.0037	0.0042	0.0032
Exactly	Mean	**0.0046**	0.0067	0.0061	0.1207	0.0060
	Std	0.0000	0.0049	0.0029	0.1046	0.0031
Exactly2	Mean	0.2340	0.2397	**0.2338**	0.2494	0.2345
	Std	0.0096	0.0027	0.0080	0.0115	0.0140
HeartEW	Mean	**0.1738**	0.1865	0.1837	0.1813	0.1883
	Std	0.0105	0.0086	0.0095	0.0106	0.0059
IonosphereEW	Mean	**0.0720**	0.0815	0.0959	0.0840	0.0873
	Std	0.0071	0.0040	0.0050	0.0067	0.0054
Lymphographyt	Mean	**0.1383**	0.1584	0.1550	0.1570	0.1507
	Std	0.0128	0.0091	0.0090	0.0105	0.0072
M-of-n	Mean	**0.0046**	0.0055	0.0065	0.0649	0.0059
	Std	0.0000	0.0012	0.0019	0.0583	0.0019
PenglungEW	Mean	**0.0833**	0.0899	0.0976	0.0908	0.0867
	Std	0.0070	0.0062	0.0064	0.0066	0.0052
SonarEW	Mean	**0.0784**	0.1012	0.0984	0.0988	0.0937
	Std	0.0130	0.0062	0.0061	0.0111	0.0052
SpectEW	Mean	**0.1565**	0.1672	0.1737	0.1662	0.1767
	Std	0.0111	0.0091	0.0056	0.0094	0.0051
Tic-tac-toe	Mean	**0.1638**	0.1790	0.1705	0.1927	0.1896
	Std	0.0045	0.0083	0.0056	0.0160	0.0021
Vote	Mean	**0.0423**	0.0483	0.0594	0.0484	0.0500
	Std	0.0056	0.0025	0.0055	0.0054	0.0048
WineEW	Mean	**0.0211**	0.0255	0.0265	0.0254	0.0242
	Std	0.0047	0.0042	0.0036	0.0049	0.0047
Zoo	Mean	0.0209	0.0217	0.0168	0.0220	**0.0136**
	Std	0.0031	0.0036	0.0039	0.0041	0.0037
AvgFitness		**0.0818**	0.0899	0.0916	0.1018	0.0903

The significance of bold means the best values.

Table 4
Results of ***AvgFR*** and their STD for ECSA and four CSA based approaches.

Datasets	Stat. measure	ECSA	ICSA1	ICSA2	CCSA	CSA
Breastcancer	Mean	**0.5556**	0.5667	0.7500	0.6111	0.6278
	Std	0.0806	0.1190	0.1074	0.1672	0.1409
BreastEW	Mean	0.4633	**0.4533**	0.7483	0.4800	0.6033
	Std	0.0865	0.0988	0.0776	0.0875	0.0830
CongressEW	Mean	0.3000	**0.2813**	0.5688	0.3563	0.4875
	Std	0.0851	0.0745	0.1215	0.1131	0.0801
Exactly	Mean	**0.4615**	0.5192	0.5192	0.6692	0.5231
	Std	0.0000	0.0491	0.0423	0.1173	0.0402
Exactly2	Mean	0.3154	**0.1269**	0.7654	0.2769	0.7500
	Std	0.3338	0.1584	0.0172	0.1547	0.0342
HeartEW	Mean	**0.5154**	0.5192	0.7000	0.5269	0.6846
	Std	0.1561	0.1270	0.0785	0.1350	0.1113
IonosphereEW	Mean	**0.3882**	0.3941	0.6309	0.4485	0.5118
	Std	0.0790	0.0584	0.1082	0.0786	0.0543
Lymphographyt	Mean	**0.4833**	0.5194	0.7472	0.5444	0.6833
	Std	0.0767	0.1008	0.0775	0.0997	0.0828
M-of-n	Mean	**0.4615**	0.5154	0.5538	0.6462	0.5385
	Std	0.0000	0.0439	0.0316	0.1099	0.0353
PenglungEW	Mean	0.4672	**0.4511**	0.7400	0.4646	0.6057
	Std	0.0221	0.0578	0.0524	0.0323	0.0372
SonarEW	Mean	**0.5083**	0.5300	0.7692	0.5000	0.6317
	Std	0.0523	0.0591	0.0493	0.0471	0.0518
SpectEW	Mean	**0.4659**	0.5000	0.7386	0.5136	0.5591
	Std	0.0779	0.1084	0.1102	0.1044	0.1124
Tic-tac-toe	Mean	0.7778	0.7611	0.7778	0.6833	**0.6111**
	Std	0.0000	0.0544	0.0000	0.0544	0.0987
Vote	Mean	0.3031	**0.2719**	0.5750	0.4625	0.4313
	Std	0.0913	0.0867	0.1539	0.1509	0.1232
WineEW	Mean	0.5808	**0.5731**	0.7346	0.5962	0.6654
	Std	0.1465	0.1073	0.0682	0.0823	0.1350
Zoo	Mean	**0.3781**	0.5000	0.7031	0.5875	0.6750
	Std	0.1773	0.1298	0.0970	0.1240	0.1006
Avgreduction		**0.4641**	0.4677	0.6889	0.5230	0.5993

The significance of bold means the best values.

measure, we have reported the mean and standard deviation for 20 independent runs for each optimizer.

Numerical results reported in these tables assess that ECSA success to hit the best results on the majority of the datasets for classification accuracy, fitness values, and the ratio of selected features. From Table 2, it is seen that the ECSA approach provides the best results on 87.50% of the datasets (fourteen datasets out of sixteen datasets). ICSA2 and CSA achieve the best results only for two datasets, while ICSA1 and CCSA give the worst results for all datasets. The reported results in Table 2 demonstrate that the embedded modifications have enhanced the exploration capacity of CSA compared to the three other variants, namely ICSA1, ICSA2 and CCSA. Hence, the adaptive AP helps CSA to balance between exploration and exploitation, to avoid premature convergence and to switch from random global exploration to a guided exploration around the global best solution. From Table 2, the ECSA gave the best average classification accuracy among other optimizers.

From Table 3, it can be observed again that ECSA outperforms the other three optimizers and obtains a significantly better average fitness measure on fourteen out of sixteen datasets. Again, ICSA2 and CSA give the best results for two datasets, while ICSA1 provides the worst results for all datasets. According to the AvgFitness row of Table 3, ECSA obtained the lowest average fitness value among the three optimizers. The modifications introduced to the original CSA help crows to explore more efficiently the search space. As a consequence, ECSA can escape from suboptimal solutions in comparison with other optimizers that are trapped in local ones.

The average of the ratio of selected features using the four approaches is reported in Table 4. From this table, we can be seen that ECSA and ICSA1 are very competitive, with values 0.4641 and 0.4677, respectively. However, ECSA selected the smallest average reduction for nine datasets, while ICSA1 has the smallest values for six datasets. Based on the numerical results recorded in Table 4, CSA gives better results for one dataset, while ICSA2 is the worst optimizer among the four others. The reason is that the new modifications have enhanced the searching competences of CSA on the majority of datasets.

In addition, the convergence curve of the five algorithms is given in Fig. 2. It can be observed that the proposed ECSA can convergence faster than the other algorithms in most of the datasets except at Exactly, Exactly2, and Zoo. However, its results are competitive with these algorithms.

Table 5
Results of Wilcoxon test for the ECSA vs. variants of CSA ($p \geq 0.05$ are in bold).

Datasets	**ICSA1**	**ICSA2**	**CCSA**	**CSA**
Breastcancer	3.14E−04	1.02E−04	2.49E−04	6.06E−03
BreastEW	3.62E−04	8.84E−05	8.84E−05	1.40E−04
CongressEW	**5.21E−02**	8.86E−05	1.03E−04	8.78E−05
Exactly	2.44E−04	1.22E−04	8.46E−05	6.10E−05
Exactly2	3.52E−02	**9.40E−01**	5.17E−04	**1.00E + 00**
HeartEW	4.49E−04	3.74E−03	6.34E−04	1.89E−04
IonosphereEW	1.88E−04	8.84E−05	8.83E−05	8.84E−05
Lymphographyt	7.28E−04	3.89E−04	5.89E−04	6.29E−04
M-of-n	2.44E−04	5.10E−05	8.49E−05	4.79E−05
PenglungEW	1.12E−02	2.19E−04	8.78E−05	2.51E−02
SonarEW	8.83E−05	1.20E−04	1.40E−04	8.86E−05
SpectEW	6.06E−03	3.90E−04	4.49E−04	1.40E−04
Tic-tac-toe	2.13E−04	1.00E−03	1.88E−04	8.75E−05
Vote	2.53E−03	8.84E−05	8.79E−05	5.36E−04
WineEW	6.62E−03	6.54E−04	5.83E−04	6.43E−02
Zoo	**2.45E−01**	2.05E−02	**4.93E−01**	9.80E−05

The significance of bold means the best values.

Table 6
Overall ranking results of the five optimizers.

Algorithms	ECSA			ICSA1			ICSA2			CCSA			CSA		
Metrics	Acc	Fit	Reduc	Acc	Fit	Reduc	Acc	Fit	Reduc	Acc	Fit	Reduc	Acc	Fit	Reduc
Breastcancer	**1**	**1**	**1**	5	4	2	2	5	5	4	2	3	3	3	4
BreastEW	**1**	**1**	2	3	3	**1**	4	5	5	2	2	3	5	4	4
CongressEW	**1**	**1**	2	2	2	**1**	5	5	5	3	3	3	4	4	4
Exactly	**1**	**1**	**1**	4	4	2.5	3	3	2.5	5	5	5	2	2	4
Exactly2	3	2	3	4	4	**1**	**1**	**1**	5	5	5	2	2	3	4
HeartEW	**1**	**1**	**1**	4	4	2	3	3	5	2	2	3	5	5	4
IonosphereEW	**1**	**1**	**1**	2	2	2	5	5	5	3	3	3	4	4	4
Lymphography	**1**	**1**	**1**	5	5	2	3	3	5	4	4	3	2	2	4
M-of-n	**1**	**1**	**1**	2	2	2	4	4	4	5	5	5	3	3	3
PenglungEW	**1**	**1**	3	3	3	**1**	5	5	5	4	4	2	2	2	4
SonarEW	**1**	**1**	2	5	5	3	3	3	5	4	4	**1**	2	2	4
SpectEW	**1**	**1**	**1**	3	3	2	4	4	5	2	2	3	5	5	4
Tic-tac-toe	**1**	**1**	4.5	3	3	3	2	2	4.5	5	5	2	4	4	**1**
Vote	**1**	**1**	2	3	2	**1**	5	5	5	2	3	4	4	4	3
WineEW	**1**	**1**	2	5	4	**1**	3	5	5	4	3	3	2	2	4
Zoo	5	3	**1**	4	4	2	2	2	5	3	5	3	**1**	**1**	4
Sum of the ranks	22	19	28.5	57	54	28.5	54	60	76	57	57	48	54	50	59
Sum of ranks	1.3750	1.1875	1.78125	3.5625	3.375	1.78125	3.3750	3.75	4.75	3.5625	3.5625	3	3.1250	3.125	3.6875
Overall ranks	69.5			139.5			190			162			159		
Final ranks	**1**			2			5			4			3		

The significance of bold means the best values.

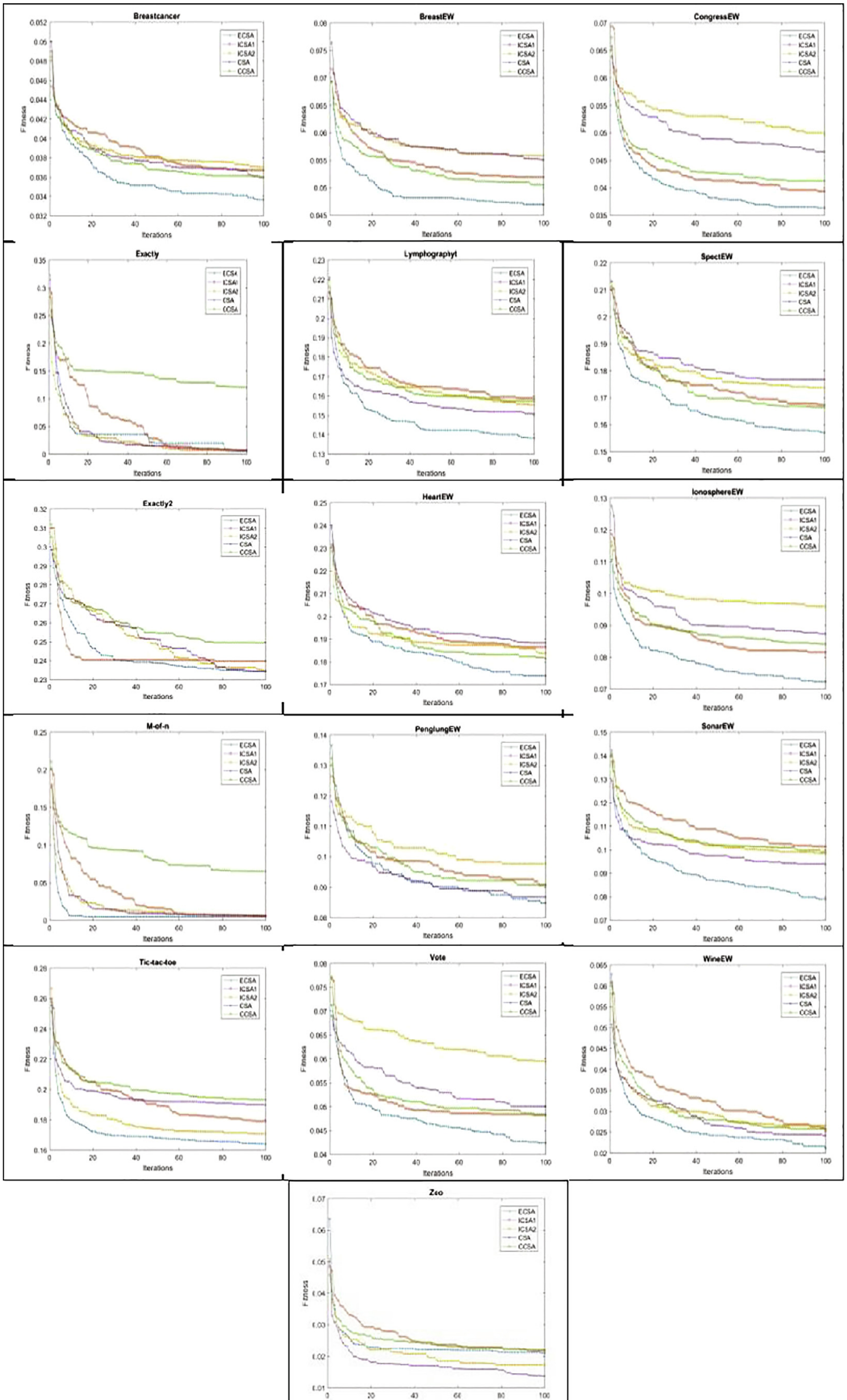

Fig. 2. Convergence curves of the ECSA, ICSA1, ICSA2, CCSA, and CSA for all datasets.

Table 7
Parameters setting for experiments.

Algorithm	Parameters	Value
GWO	α	2, 0
GSA	G0	100
GSA	α	20
ABC	Trail	100
	modification rate (MR)	0.8
PSOGWO	c1, c2, c3	0.5
	w	0.5 + rand()/2
SCA	α	2
DE	F, CR	0.3 0.7
BOA	c	0.01
	α	0.1
	p	0.8
BDA	A	[−0.1,0,1]
	s (Seperation weight)	2*rand()*A
	a (Alignment weight)	2*rand()*A
	c (Cohesion weight)	2*rand()*A
	f (Food attraction weight)	2*rand()
	e (Enemy distraction weight)	e = A
SCADE	α	2.0
	F, CR	0.3, 0.7
SCAGWO	α	2.0
SSA	c_1	Decreases linearly from 2 to 0

To further analyse the results, the non-parametric Wilcoxon test is used (Derrac, García, Molina, & Herrera, 2011), which determines if there is a significant difference between the proposed ECSA as a control group and the other methods (i.e., ICSA1, ICSA2, CCSA, and CSA). According to the results of the Wilcoxon test given in Table 5, it can be observed that there are significant differences between the ECSA and other methods among all tested datasets except at two datasets. The first dataset is Exactly2, and it can be noticed that the accuracy of the propose ECSA is nearly the same as traditional and the ICSA2. The same observation can be noticed for the dataset Zoo, however, this time no significant difference with the ICSA1, and CCSA.

Moreover, it is clear from Table 6 that ECSA obtains the first place among other CSA versions regarding the accuracy, fitness, and feature reduction metrics. There is a great difference between ECSA and the four variants of CSA in terms of accuracy and Fitness measures, but in terms of feature reduction, ECSA and ICSA1 have the same performance. According to the final ranks, ECSA and ICSA1 are in the first and second places, while CSA and CCSA obtain the third and four places, whereas, the last place is for ICSA2.

5.4. Experimental series 2: Comparison with other MH methods

In the second experimental series, the performance of ECSA is compared with eight metaheuristics that have been widely used to solve the feature selection problem. The metaheuristics used for the comparison in this work are binary Bufferly Optimization algorithm (BOA) (Arora & Singh, 2019), Binary Gravitational search algorithm (BGSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009), Differential Evolution (DE) (Storn & Price, 1997), Artificial bee colony (ABC) (Karaboga & Basturk, 2008), SCADE (Elaziz, Ewees, Oliva, Duan & Xiong, 2017), PSOGWO (Singh & Singh, 2017a), SCAGWO (Singh & Singh, 2017b), SSA (Mirjalili et al., 2017) and SCA (Mirjalili, 2016). These algorithms have been established their performance as a feature selection method in the recent literature. In addition, we use the KNN algorithm with all the features set. The KNN is a simple classifier that searches the K closest labeled data to classify unlabeled ones based on some distance measure. In KNN, we adopt the Euclidean distance with K = 5 for all experiments. The KNN method is used to assess the impact of selection of optimal subset features on the accuracy of the classifier. Table 7 tabulates the parameter values used in this experiment for all these methods. The parameters of each optimizer used in the paper for comparison are fixed according to their corresponding papers.

In Tables 8–11, we tabulate the average and standard values for accuracy, fitness, the ratio of selected features and running time for all techniques, respectively. Table 8 records the results of average and standard values of classification accuracy for all algorithms for each dataset. We can observe from Table 8 that ECSA outperforms all other approaches on 93.75% of the datasets. From the row AvgAccuracy, ECSA gave the highest average result followed by PSOGWO with values 92.20% and 91.68%, respectively. The worst results are those obtained without applying feature selection (it given by the ALL). In terms of fitness values, we can observe from Table 9 and Fig. 3 that ECSA is ranked in the first place and has surpassed other metaheuristics on fourteen datasets. BOA and ABC gave better the smallest value of fitness for one dataset. All these tabulated results demonstrated the merits of the proposed ECSA algorithm for dealing with feature selection problem.

In addition, Table 10 showed the ratio of selected features for all algorithms. We can observe from this Table that ECSA and PSOGWO are competitive since ECSA performs better on nine datasets with average ratio value 0.4641, while PSOGWO has a better performance for eight datasets with average ratio value 0.4677.

From Table 11, we can see that ECSA presents an acceptable computational time in comparison with other metaheuristics. In general, by analyzing the average of CPU time(s) overall tested dataset as in Fig. 4, it has been observed that the average of ABC and ECSA nearly the same. However, the proposed ECSA has the smallest CPU time (s) at five datasets, followed by ABC and BDA with four and three datasets, respectively. Each of BOA, PSOGWO, SSA, and SCA has the smallest CPU time (s) at one dataset. The use of the local neighborhood search in ECSA caused an increase in the computation time; however, experimental results verify the benefits of the additional cost caused by the modifications in the accuracy of classification.

Moreover, the performance of ECSA is validated statistically by using the nonparametric Wilcoxon's rank-sum test (Derrac et al., 2011). From Table 12, we can see that ECSA is statistically better than other optimizers in dealing with the majority of the datasets, which verifies the efficacy of the proposed modifications introduced to the original CSA algorithm.

5.5. Experimental series 3: Comparison with other optimizers from literature

In this section, we will assess the performance of the proposed ECSA by comparing it with other FS methods based on different MH algorithms. These methods include Whale Optimization algorithm and two variants of it (WOA, WOAT, and WOAR), two variants of binary Grey wolf optimization (bGWO1 and bGWO2), Binary Bat Algorithm (BBA), Genetic algorithm (GA), Particle swarm optimization (PSO), Binary Grasshopper optimization (BGOA), Biogeography-based optimization (BBO), Satin Bird Optimizer (SBO), Enhanced GWO (EGWO) and Augmented GWO (AGWO). The results of WOA, WOAT, and WOAR are from (Mafarja & Mirjalili, 2018), those of GA, PSO, bGWO1, and bGWO2 are from (Emary, Zawbaa, & Hassanien, 2016). The results of BBA and BGOA are from (Mafarja et al., 2019), and those of BBO, SBO, EGWO, and AGWO are from (Arora, Singh, Sharma, Sharma, & Anand 2019).

Table 13 shows the accuracy of the proposed ECSA against the other state-of-the-art methods. From these results, it can be concluded that the proposed ECSA achieves the best accuracy at eight datasets. While bGWO2 and BGOA have the best at three datasets, also, each of the WOA, WOAT, and GA has the best accuracy at only one dataset.

Table 8
Comparison between ECSA and other methods in terms of average and std classification accuracy.

Datasets	Measure	ECSA	BOA	BGSA	BDA	DE	ABC	SCADE	PSOGWO	SCAGWO	ALL	SSA	SCA
Breastcancer	Mean	**0.9716**	0.9696	0.9687	0.9642	0.9700	0.9715	0.9692	0.9708	0.9699	0.9506	0.9700	0.9692
	Std	0.0019	0.0016	0.0020	0.0036	0.0017	0.0018	0.0012	0.0014	0.0020	0.0037	0.0021	0.0014
BreastEW	Mean	**0.9576**	0.9502	0.9495	0.9433	0.9522	0.9517	0.9518	0.9557	0.9506	0.9294	0.9518	0.9487
	Std	0.0035	0.0033	0.0031	0.0042	0.0022	0.0033	0.0028	0.0028	0.0023	0.0056	0.0028	0.0031
CongressEW	Mean	**0.9663**	0.9566	0.9529	0.9520	0.9589	0.9631	0.9584	0.9657	0.9568	0.9149	0.9584	0.9543
	Std	0.0043	0.0036	0.0041	0.0034	0.0029	0.0044	0.0029	0.0033	0.0038	0.0076	0.0032	0.0029
Exactly	Mean	**1.0000**	0.9959	0.9970	0.8280	**1.0000**	**1.0000**	0.9996	**1.0000**	0.9886	0.7387	0.9997	0.9973
	Std	0.0000	0.0066	0.0065	0.0913	0.0000	0.0000	0.0020	0.0000	0.0165	0.0125	0.0010	0.0050
Exactly2	Mean	**0.7669**	0.7678	0.7568	0.7513	0.7628	0.7598	0.7586	0.7648	0.7494	0.6621	0.7589	0.7621
	Std	0.0129	0.0178	0.0071	0.0139	0.0138	0.0173	0.0120	0.0093	0.0144	0.0096	0.0127	0.0195
HeartEW	Mean	**0.8296**	0.8165	0.7965	0.7961	0.8246	0.8278	0.8191	0.8237	0.8148	0.7387	0.8178	0.8159
	Std	0.0101	0.0058	0.0104	0.0090	0.0076	0.0084	0.0062	0.0063	0.0071	0.0190	0.0090	0.0076
IonosphereEW	Mean	**0.9312**	0.9137	0.9105	0.9093	0.9194	0.9261	0.9157	0.9286	0.9121	0.5580	0.9151	0.9121
	Std	0.0069	0.0032	0.0081	0.0067	0.0048	0.0083	0.0053	0.0048	0.0046	0.1035	0.0057	0.0053
Lymphographyt	Mean	**0.8652**	0.8544	0.8355	0.8392	0.8568	0.8574	0.8507	0.8568	0.8480	0.7480	0.8486	0.8497
	Std	0.0127	0.0081	0.0110	0.0134	0.0075	0.0105	0.0103	0.0092	0.0102	0.0128	0.0064	0.0090
M-of-n	Mean	**1.0000**	0.9997	0.9975	0.9556	**1.0000**	**1.0000**	0.9998	1.0000	0.9938	0.8335	0.9995	0.9988
	Std	0.0000	0.0008	0.0056	0.0502	0.0002	0.0000	0.0009	0.0000	0.0076	0.0100	0.0013	0.0020
PenglungEW	Mean	**0.9205**	0.9158	0.9041	0.9055	0.9144	0.9062	0.9116	0.9171	0.9089	0.8274	0.9116	0.9171
	Std	0.0072	0.0067	0.0089	0.0076	0.0087	0.0102	0.0070	0.0070	0.0067	0.0253	0.0083	0.0054
SonarEW	Mean	**0.9260**	0.9108	0.8940	0.8964	0.9147	0.9041	0.9123	0.9135	0.9075	0.8510	0.9099	0.9103
	Std	0.0131	0.0055	0.0099	0.0065	0.0085	0.0107	0.0062	0.0080	0.0060	0.0101	0.0084	0.0074
SpectEW	Mean	**0.8466**	0.8260	0.8185	0.8257	0.8354	0.8333	0.8311	0.8416	0.8330	0.7328	0.8333	0.8219
	Std	0.0111	0.0048	0.0072	0.0087	0.0069	0.0094	0.0066	0.0070	0.0046	0.0282	0.0061	0.0055
Tic-tac-toe	Mean	**0.8424**	0.8143	0.8118	0.8009	0.8354	0.8435	0.8337	0.8341	0.8309	0.7532	0.8308	0.8121
	Std	0.0045	0.0024	0.0031	0.0079	0.0041	0.0044	0.0061	0.0045	0.0068	0.0136	0.0041	0.0033
Vote	Mean	**0.9603**	0.9522	0.9492	0.9497	0.9493	0.9537	0.9495	0.9578	0.9465	0.8995	0.9483	0.9505
	Std	0.0054	0.0033	0.0039	0.0055	0.0038	0.0037	0.0049	0.0039	0.0037	0.0087	0.0047	0.0036
WineEW	Mean	**0.9846**	0.9809	0.9725	0.9697	0.9798	0.9820	0.9792	0.9823	0.9789	0.9494	0.9775	0.9826
	Std	0.0040	0.0038	0.0065	0.0062	0.0038	0.0039	0.0037	0.0027	0.0036	0.0095	0.0032	0.0040
Zoo	Mean	0.9827	0.9916	0.9817	0.9787	0.9891	0.9837	0.9851	0.9851	0.9861	0.9574	0.9881	0.9921
	Std	0.0044	0.0036	0.0048	0.0058	0.0030	0.0048	0.0051	0.0051	0.0050	0.0144	0.0052	0.0041
Avg Accuracy		**0.9220**	0.9135	0.9060	0.8916	0.9164	0.9165	0.9141	0.9186	0.9110	0.8153	0.9137	0.9122

The significance of bold means the best values.

Table 9
Comparison between ECSA and other methods in terms of average and std fitness.

Datasets	Stat. measure	ECSA	BOA	BGSA	BDA	DE	ABC	SCADE	PSOGWO	SCAGWO	SSA	SCA
Breastcancer	Mean	**0.0337**	0.0364	0.0371	0.0411	0.0360	0.0345	0.0366	0.0346	0.0371	0.0362	0.0372
	Std	0.0019	0.0014	0.0017	0.0031	0.0016	0.0017	0.0016	0.0014	0.0016	0.0021	0.0010
BreastEW	Mean	**0.0467**	0.0553	0.0551	0.0610	0.0533	0.0529	0.0538	0.0486	0.0550	0.0540	0.0566
	Std	0.0038	0.0036	0.0032	0.0039	0.0023	0.0033	0.0029	0.0029	0.0022	0.0030	0.0031
CongressEW	Mean	**0.0363**	0.0484	0.0500	0.0517	0.0452	0.0406	0.0458	0.0369	0.0475	0.0454	0.0499
	Std	0.0045	0.0037	0.0045	0.0032	0.0028	0.0049	0.0028	0.0033	0.0040	0.0037	0.0026
Exactly	Mean	**0.0046**	0.0097	0.0083	0.1773	0.0052	0.0046	0.0057	0.0046	0.0171	0.0056	0.0082
	Std	0.0000	0.0069	0.0069	0.0902	0.0004	0.0000	0.0022	0.0000	0.0167	0.0011	0.0052
Exactly2	Mean	**0.2340**	0.2372	0.2427	0.2493	0.2421	0.2426	0.2462	0.2368	0.2538	0.2455	0.2424
	Std	0.0096	0.0171	0.0068	0.0127	0.0135	0.0165	0.0120	0.0067	0.0130	0.0121	0.0190
HeartEW	Mean	**0.1738**	0.1885	0.2076	0.2082	0.1801	0.1764	0.1858	0.1800	0.1897	0.1866	0.1891
	Std	0.0105	0.0058	0.0104	0.0084	0.0078	0.0089	0.0063	0.0059	0.0071	0.0091	0.0077
IonosphereEW	Mean	**0.0720**	0.0908	0.0931	0.0940	0.0851	0.0774	0.0887	0.0741	0.0924	0.0894	0.0922
	Std	0.0071	0.0030	0.0082	0.0067	0.0048	0.0087	0.0052	0.0050	0.0047	0.0055	0.0052
Lymphographyt	Mean	**0.1383**	0.1503	0.1685	0.1642	0.1482	0.1467	0.1544	0.1469	0.1568	0.1561	0.1551
	Std	0.0128	0.0078	0.0103	0.0138	0.0075	0.0103	0.0098	0.0091	0.0097	0.0065	0.0084
M-of-n	Mean	**0.0046**	0.0057	0.0081	0.0506	0.0050	0.0046	0.0054	0.0046	0.0121	0.0059	0.0067
	Std	0.0000	0.0011	0.0059	0.0502	0.0005	0.0000	0.0012	0.0000	0.0079	0.0015	0.0024
PenglungEW	Mean	**0.0833**	0.0894	0.0998	0.0984	0.0910	0.0979	0.0937	0.0866	0.0964	0.0937	0.0882
	Std	0.0070	0.0066	0.0087	0.0072	0.0084	0.0101	0.0068	0.0069	0.0065	0.0080	0.0051
SonarEW	Mean	**0.0784**	0.0944	0.1100	0.1077	0.0910	0.1003	0.0932	0.0905	0.0982	0.0954	0.0953
	Std	0.0130	0.0053	0.0101	0.0066	0.0085	0.0106	0.0063	0.0079	0.0058	0.0082	0.0074
SpectEW	Mean	**0.1565**	0.1782	0.1844	0.1768	0.1689	0.1698	0.1733	0.1616	0.1718	0.1713	0.1827
	Std	0.0111	0.0048	0.0071	0.0086	0.0064	0.0092	0.0067	0.0068	0.0051	0.0064	0.0053
Tic-tac-toe	Mean	0.1638	0.1900	0.1920	0.2036	0.1707	0.1627	0.1724	0.1719	0.1751	0.1752	0.1928
	Std	0.0045	0.0019	0.0034	0.0081	0.0041	0.0043	0.0060	0.0041	0.0067	0.0041	0.0031
Vote	Mean	**0.0423**	0.0515	0.0539	0.0534	0.0551	0.0497	0.0545	0.0442	0.0580	0.0559	0.0533
	Std	0.0056	0.0030	0.0042	0.0054	0.0039	0.0041	0.0052	0.0041	0.0036	0.0051	0.0037
WineEW	Mean	**0.0211**	0.0254	0.0328	0.0356	0.0260	0.0233	0.0267	0.0224	0.0270	0.0286	0.0241
	Std	0.0047	0.0035	0.0064	0.0057	0.0038	0.0040	0.0034	0.0026	0.0034	0.0029	0.0045
Zoo	Mean	0.0209	0.0150	0.0243	0.0272	0.0177	0.0211	0.0208	0.0196	0.0206	0.0185	0.0146
	Std	**0.0031**	0.0029	0.0045	0.0054	0.0024	0.0037	0.0039	0.0035	0.0038	0.0041	0.0034
Avg fitness		**0.0818**	0.0916	0.0980	0.1125	0.0888	0.0878	0.0911	0.0852	0.0943	0.0915	0.0930

The significance of bold means the best values.

Table 10
Comparison between ECSA and other methods in terms of average and std selection reduction.

Datasets	Stat. measure	ECSA	BOA	BGSA	BDA	DE	ABC	SCADE	PSOGWO	SCAGWO	SSA	SCA
Breastcancer	Mean	**0.5556**	0.6333	0.6111	0.6722	0.6333	0.6222	0.6111	0.5722	0.7333	0.6500	0.6722
	Std	0.0806	0.0961	0.0987	0.0917	0.1087	0.0981	0.1111	0.0828	0.1320	0.1037	0.1167
BreastEW	Mean	**0.4633**	0.6017	0.5033	0.6100	0.5983	0.5083	0.6100	0.4800	0.6100	0.6217	0.5750
	Std	0.0865	0.1017	0.0809	0.0845	0.0729	0.0858	0.0765	0.0775	0.0918	0.0759	0.0764
CongressEW	Mean	**0.3000**	0.5344	0.3375	0.4844	0.4469	0.4094	0.4625	**0.3000**	0.4750	0.4219	0.4625
	Std	0.0851	0.1191	0.1320	0.1264	0.0913	0.1173	0.0821	0.0851	0.0769	0.0926	0.1396
Exactly	Mean	**0.4615**	0.5577	0.5269	0.5308	0.5154	**0.4615**	0.5231	**0.4615**	0.5808	0.5231	0.5462
	Std	0.0000	0.0423	0.0516	0.0606	0.0362	0.0000	0.0402	0.0000	0.0465	0.0316	0.0344
Exactly2	Mean	0.3154	0.7308	**0.1962**	0.6808	0.7231	0.4846	0.7192	0.3923	0.5654	0.6769	0.6846
	Std	0.3338	0.1210	0.1941	0.1969	0.1207	0.3088	0.1544	0.3502	0.2397	0.1758	0.1883
HeartEW	Mean	**0.5154**	0.6808	0.6115	0.6692	0.6462	0.5923	0.6654	0.5500	0.6346	0.6154	0.6885
	Std	0.1561	0.0718	0.0682	0.0793	0.0914	0.1200	0.1255	0.1037	0.1055	0.1171	0.0768
IonosphereEW	Mean	0.3882	0.5338	0.4574	0.5221	0.5265	0.4176	0.5206	**0.3412**	0.5353	0.5353	0.5176
	Std	0.0790	0.0649	0.0544	0.0700	0.0660	0.0802	0.0782	0.0818	0.0672	0.0749	0.0882
Lymphographyt	Mean	**0.4833**	0.6094	0.5639	0.6278	0.6389	0.5583	0.6528	0.5083	0.6250	0.6306	0.6278
	Std	0.0767	0.1005	0.1100	0.0788	0.0979	0.1293	0.1018	0.0924	0.1033	0.0941	0.0903
M-of-n	Mean	**0.4615**	0.5423	0.5538	0.5308	0.5000	**0.4615**	0.5231	**0.4615**	0.5962	0.5385	0.5462
	Std	0.0000	0.0393	0.0535	0.0552	0.0395	0.0000	0.0402	0.0000	0.0423	0.0353	0.0425
PenglungEW	Mean	0.4672	0.6022	0.4842	0.6095	0.6237	0.5015	0.6203	**0.4546**	0.6258	0.6238	0.6112
	Std	0.0221	0.0302	0.0248	0.0362	0.0350	0.0281	0.0416	0.0273	0.0408	0.0484	0.0384
SonarEW	Mean	0.5083	0.6065	0.5092	0.6283	0.6483	0.5333	0.6325	**0.4842**	0.6550	0.6200	0.6483
	Std	0.0523	0.0641	0.0615	0.0684	0.0623	0.0741	0.0579	0.0611	0.0565	0.0400	0.0575
SpectEW	Mean	**0.4659**	0.5977	0.4705	0.6159	0.5909	0.4795	0.6045	0.4795	0.6432	0.6273	0.6432
	Std	0.0779	0.0811	0.0947	0.0983	0.1000	0.0983	0.0645	0.1166	0.1173	0.0855	0.0959
Tic-tac-toe	Mean	0.7778	0.6111	**0.5667**	0.6444	0.7778	0.7778	0.7778	0.7611	0.7778	0.7778	0.6778
	Std	0.0000	0.0987	0.0497	0.1117	0.0000	0.0000	0.0000	0.0544	0.0000	0.0000	0.1134
Vote	Mean	0.3031	0.4210	0.3594	0.5031	0.4906	0.3875	0.4500	**0.2500**	0.5063	0.4719	0.4313
	Std	0.0913	0.1440	0.1280	0.1258	0.1116	0.1084	0.1276	0.0608	0.1359	0.1455	0.1145
WineEW	Mean	0.5808	0.6500	0.5500	0.7308	0.6000	0.5538	0.6077	**0.4923**	0.6115	0.6308	0.6885
	Std	0.1465	0.1157	0.1067	0.1608	0.0886	0.1047	0.1342	0.0765	0.1073	0.0920	0.1129
Zoo	Mean	**0.3781**	0.6688	0.6156	0.6469	0.6906	0.4906	0.6094	0.4938	0.6906	0.6781	0.6719
	Std	0.1773	0.0931	0.1078	0.1097	0.0872	0.1480	0.1403	0.1868	0.1537	0.1423	0.0904
Avg reduction		**0.4641**	0.5988	0.4948	0.6067	0.6032	0.5150	0.5994	0.4677	0.6166	0.6058	0.6027

The significance of bold means the best values.

Table 11
Average CPU time (s) of ECSA compared and other metaheuristics.

Datasets	ECSA	BOA	BGSA	BDA	DE	ABC	SCADE	PSOGWO	SCAGWO	SSA	SCA
Breastcancer	**14.41**	16	15.41	14.89	14.86	14.61	14.85	14.7	15.47	14.88	14.59
BreastEW	13.98	14.98	14.35	14.13	14.57	**13.84**	14.47	14.24	15.31	14.29	14.29
CongressEW	12.7	12.73	12.95	12.99	12.86	12.7	12.73	12.87	12.88	12.66	**12.65**
Exactly	17.82	19.36	17.68	**17.07**	18.52	17.62	18.69	17.65	18.74	18.28	18.59
Exactly2	18.59	19.82	18.13	19.26	19.64	18.81	19.4	**17.02**	19.6	18.88	18.89
HeartEW	**11.44**	11.66	12.02	11.72	11.64	11.55	11.64	11.81	11.85	11.54	11.53
IonosphereEW	11.76	12.34	12.91	12.01	12.15	**11.6**	12.13	12.32	12.42	11.94	11.91
Lymphographyt	**10.54**	10.76	12.18	11.11	10.65	10.62	10.66	11.12	10.83	10.68	10.6
M-of-n	17.63	18.95	17.53	**16.93**	18.44	17.23	18.37	17.49	18.75	18.2	18.02
PenglungEW	11.9	13.29	12.1	12.24	12.15	**11.65**	12.19	17.01	17	12.2	12.12
SonarEW	11.05	11.55	11.85	11.19	11.37	**10.95**	11.44	11.54	11.8	11.2	11.13
SpectEW	10.89	11.13	12	11.44	11.07	11.05	11.11	11.62	11.31	**10.89**	10.89
Tic-tac-toe	16.64	17.53	16.04	**15.11**	19.55	17.13	17.34	16.84	17.41	17.06	16.46
Vote	11.61	**11.56**	12.65	12.87	11.71	11.64	11.73	11.97	11.78	11.59	11.55
WineEW	**10.87**	11.14	11.73	11.15	11.07	10.94	11.13	11.2	11.23	10.96	10.96
Zoo	**10.87**	11.44	12.02	11.48	11.03	10.93	11.09	11.27	11.37	10.98	11.06

From all of the previously discussed results, it can be observed the high performance of the proposed ECSA as a feature selection method. This achieved by enhancing CSA where the value of the AP parameter changes dynamically based on adaptive ranking strategy. The random choice of the crow to follow is replaced by a selection strategy from a local neighborhood. We also hybrid the CSA with a new global search strategy to improve the exploration process. So, the difference between ECSA and three improved CSA versions is: In (Shi & al. 2017), authors used a roulette wheel selection mechanism, which is based on a proportional probability to select the crow to follow this algorithm called ICSA1. However, this strategy can drive the population to an early convergence. In the case of the individuals in the population have very close fitness values; this leads to individuals with almost equal fitness values having approximately the same probability of being selected a crow to follow. To avoid this drawback, we propose in this paper another strategy where each crow chooses the crow to follow from its local and small neighborhood. Inspired by the particle swarm optimization (PSO), we use the ring topology for the neighborhood since many papers proved that particle swarm optimization (PSO)-based schemes adopting a ring topology for local neighborhood definition are capable of discovering multiple local/-global optima. The ring topology with a few connected neighbors assures a greater exploration and makes ECSA less prone to being

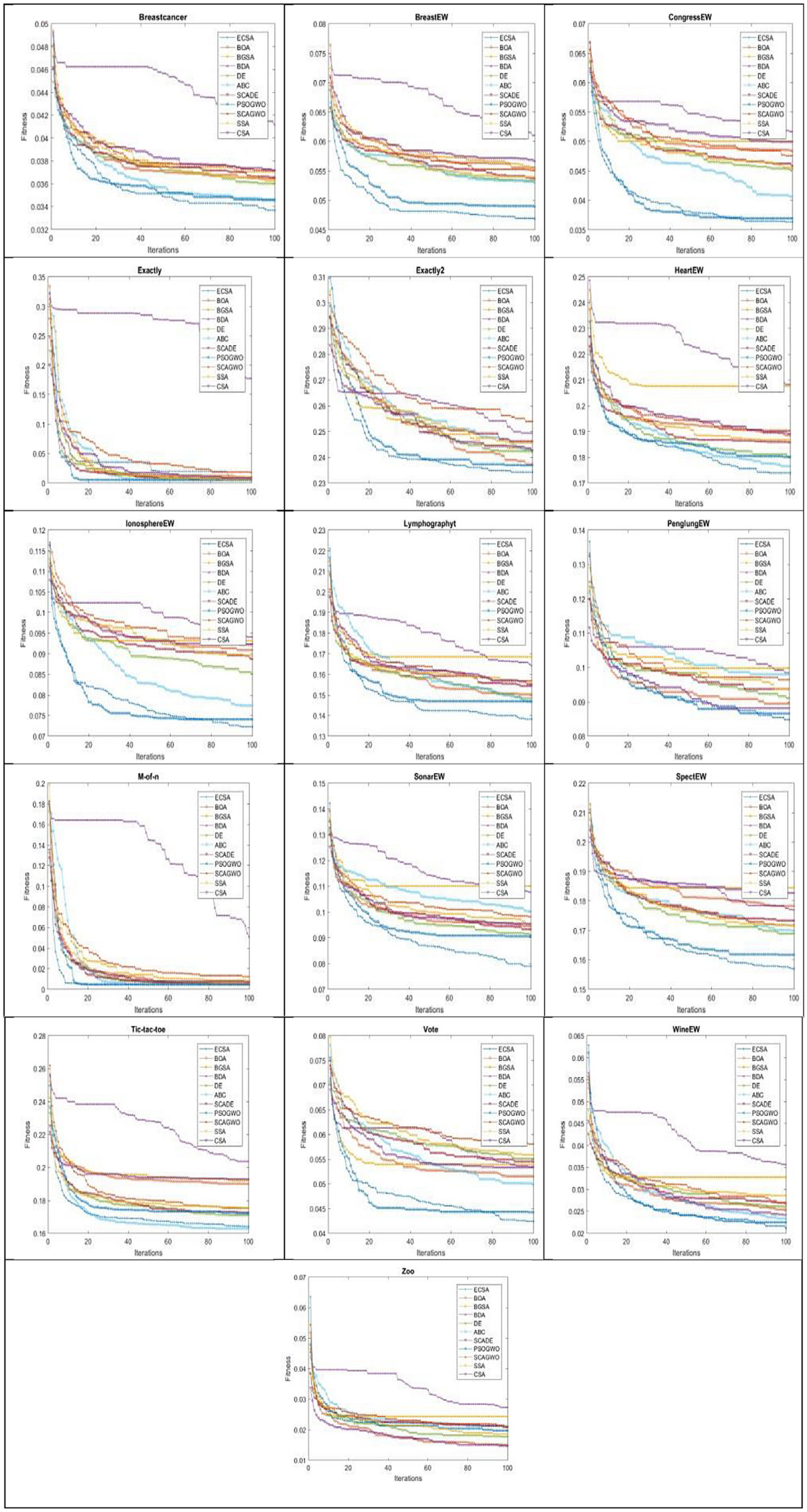

Fig. 3. Convergence curves of all algorithms for all datasets.

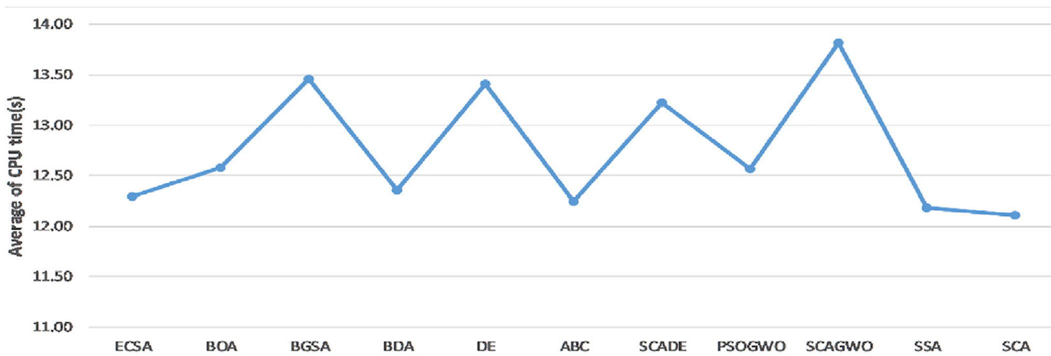

Fig. 4. Average of CPU time (s) for each algorithm.

Table 12
Results of the Wilcoxon test for ECSA vs. other MH methods ($p \geq 0.05$ are in bold).

Datasets	BOA	BGSA	BDA	DE	ABC	SCADE	PSOGWO	SCAGWO	SSA	SCA
Breastcancer	8.65E−05	3.25E−04	1.03E−04	9.98E−04	**1.27E−01**	4.15E−04	3.18E−02	2.49E−04	1.71E−03	1.30E−04
BreastEW	8.83E−05	1.03E−04	8.86E−05	2.92E−04	1.75E−04	1.40E−04	3.97E−02	8.84E−05	4.49E−04	8.86E−05
CongressEW	8.77E−05	8.79E−05	8.84E−05	1.31E−04	3.75E−03	1.32E−04	**8.25E−01**	1.03E−04	1.03E−04	8.83E−01
Exactly	7.74E−02	1.22E+00	8.84E−01	1.22E+00	**1.00E+04**	6.10E−01	**1.00E+04**	8.46E−01	1.76E+00	9.11E−01
Exactly2	5.10E−01	3.75E+01	1.59E+01	**5.22E+02**	**8.40E+02**	2.54E+01	**4.55E+03**	5.17E+00	1.16E+01	4.79E+02
HeartEW	8.78E−01	1.03E+00	8.84E−01	**6.20E+02**	**4.56E+03**	1.32E+01	2.67E+02	6.34E+00	2.99E+01	8.81E−01
IonosphereEW	8.83E−01	1.03E+00	8.84E−01	8.86E−01	**1.04E+03**	8.86E−01	**3.80E+03**	8.83E−01	2.19E+00	1.20E+00
Lymphographyt	8.78E−01	1.40E+00	5.13E+00	1.41E+02	**6.45E+02**	2.06E+01	1.67E+02	5.90E+00	8.83E+00	4.62E+00
M-of-n	7.74E−02	1.58E+00	8.79E−01	1.95E+01	**1.00E+04**	6.10E−01	**1.00E+04**	8.49E−01	8.05E−01	1.30E+00
PenglungEW	8.82E−01	8.83E−01	1.63E+00	2.20E+01	8.83E−01	1.03E+00	**6.54E+03**	8.78E−01	3.59E+01	1.37E+02
SonarEW	8.86E−01	1.20E+00	1.03E+00	5.73E+01	1.40E+00	2.20E+01	3.59E+01	1.40E+00	1.02E+01	1.32E+01
SpectEW	8.83E−01	8.84E−01	1.03E+00	1.71E+01	2.82E+01	1.20E+00	8.59E+02	4.49E+00	1.62E+00	8.86E−01
Tic-tac-toe	8.52E−01	8.73E−01	8.78E−01	3.31E+00	**7.08E+03**	1.36E+00	3.25E+00	1.88E+00	8.54E−01	8.81E−01
Vote	8.82E−01	1.63E+00	1.55E+00	1.89E+00	2.17E+00	2.93E+00	**3.05E+03**	8.79E−01	1.11E+00	1.20E+00
WineEW	8.67E−01	2.53E+00	1.31E+00	7.29E+01	**2.35E+03**	1.71E+01	**3.20E+03**	5.83E+00	1.30E+00	**8.57E+02**
Zoo	5.75E−01	1.32E+02	1.49E+01	3.11E+01	**6.31E+03**	**8.10E+03**	**3.03E+03**	**4.93E+03**	**5.65E+02**	2.13E+00

The significance of bold means the best values.

Table 13
Classification accuracies of ECSA compared with other optimizers from literature.

Datasets	ECSA	WOA	WOAT	WOAR	bGWO1	bGWO2	GA	PSO	BBA	BGOA	BBO	SBO	EGWO	AGWO
Breastcancer	0.972	0.9571	0.9590	0.9576	0.976	**0.975**	0.968	0.967	0.937	0.969	0.9623	0.9674	0.9617	0.9606
BreastEW	0.958	0.9553	0.9498	0.9507	0.924	0.935	0.939	0.933	0.931	**0.960**	0.9453	0.9425	0.9474	0.9340
CongressEW	**0.966**	0.9296	0.9147	0.9106	0.708	0.776	0.674	0.688	0.872	0.953	0.9367	0.9505	0.9431	0.9358
Exactly	**1.000**	0.7576	0.7396	0.7633	0.745	0.750	0.746	0.730	0.610	0.946	0.7540	0.7340	0.7536	0.7576
Exactly2	0.767	0.6985	0.6994	0.6907	0.776	0.776	**0.780**	0.787	0.628	0.760	0.6924	0.7096	0.6988	0.6956
HeartEW	**0.830**	0.7633	0.7652	0.7633	0.744	0.700	0.696	0.744	0.754	0.826	0.7822	0.7926	0.7615	0.7970
IonosphereEW	0.931	0.8901	0.8844	0.8801	0.908	**0.963**	0.861	0.921	0.877	0.883	0.8807	0.8989	0.8636	0.8932
Lymphographyt	0.865	**0.9151**	0.8965	0.9018	0.600	0.584	0.584	0.584	0.701	0.815	0.8000	0.8182	0.7663	0.7919
M-of-n	**1.000**	0.7858	0.7786	0.7595	0.731	0.729	0.754	0.737	0.722	0.979	0.8804	0.8632	0.8704	0.8780
PenglungEW	**0.921**	0.8540	0.8389	0.8603	0.820	0.822	0.793	0.822	0.795	0.861	0.8162	0.8432	0.7568	0.8541
SonarEW	0.926	0.7297	0.7365	0.7122	0.935	**0.938**	0.932	0.928	0.844	0.895	0.8712	0.8942	0.8615	0.8827
SpectEW	0.847	0.8543	**0.8611**	0.8572	0.807	0.834	0.814	0.819	0.800	0.803	0.7985	0.7985	0.8045	0.8134
Tic-tac-toe	0.842	0.7877	0.7922	0.7787	0.728	0.727	0.719	0.735	0.665	**0.951**	0.7683	0.7683	0.7712	0.7628
Vote	**0.960**	0.7511	0.7363	0.7398	0.912	0.920	0.904	0.904	0.851	0.729	0.9173	0.9347	0.9027	0.9200
WineEW	**0.985**	0.9387	0.9350	0.9323	0.930	0.920	0.937	0.933	0.919	0.979	0.9663	0.9685	0.9663	0.9573
Zoo	0.983	0.7127	0.7101	0.7121	0.879	0.879	0.855	0.861	0.874	**0.990**	0.9373	0.9686	0.9686	0.9686
Ranks	**2.25**	7.1875	7.9375	8.1563	8.5313	7.8125	9.5313	8.7813	12.0625	4.4375	7.4688	6.1875	8.0313	6.625
Sum of ranks	**36**	115	127	130.5	136. 5	125	152.5	140.5	193	171	119.5	99	128.5	106

The significance of bold means the best values.

trapped in a local minimum. In addition, the local neighborhood is dynamic and changes each iteration, which is beneficial for introducing more diversity in the population.

In (Zamani et al., 2019), the authors proposed CCSA, which depends on a new updating position strategy by comparing the fitness value of the neighborhood with the non-neighborhood crows. The neighborhood is a spatial topology by computing the closest surrounding neighbors for each crow. However, this neighborhood is constructed from the closest crows (in terms of Euclidean distance). In addition, the local neighborhood is calculated each iteration for each crow, which increases the computational time in this paper. Experimental results demonstrate that the ring topology is more robust and improves the performance of the classical CSA better.

In (Díaz et al., 2018), the authors proposed a dynamic Awareness Probability based on the fitness of each crow (this version of

CSA named ICSA2). In this work, we investigate the use of a ranking based probability, which is, in general, more efficient than a fitness-based probability, especially when the fitness values for some crows are very close. In ECSA, a dynamic AP is calculated by using the ranking strategy, which is able to maintain the diversification and avoiding early convergence. The AP assigned to each individual depends only on its position in the individual rank and not on its fitness value. The results demonstrate the high performance of the proposed method in terms of accuracy and robustness against the classical CSA and the three proposed improved versions of CSA. However, this high quality of the proposed ECSA still requires some improvements since it has some limitations, such as the size of the neighborhood used during the local search and the impact of the binarization function used to move from the continuous space to a binary one.

6. Conclusion

In this work, a modified version of CSA is proposed as a feature selection approach. The proposed ECSA depends on using the dynamic local neighborhood to guide the local search of crows and a novel global search strategy to increase the global exploration capability of the crow. The performance of the ECSA is compared with other variants of the traditional CSA using different UCI datasets. According to the results, it has been found the proposed ECSA outperforms the other variants of CSA. Moreover, we compared the ECSA with a set of eleven well-known feature selection methods implemented in this study. From the comparison study between these methods and the proposed ECSA, it has been concluded that the ECSA outperforms the other algorithms in thirteen datasets, while its results at the rest dataset are competitive with other methods. To further prove the high quality of the proposed ECSA, its results are compared with other FS methods from the literature that used the same protocol to perform the experiments. From this comparison, it noticed that the ECSA still provides high results than other FS methods.

According to the behaviors of the proposed ECSA, it can apply it in future works to other applications, including cloud computing, image segmentation, and engineering applications. Finally, applying our novel algorithm (ECSA) to other domains, and using other FS models (e.g., filter approach using rough set theory), can be explored as a future extension of this work. In addition, the proposed method can be enhanced by reducing the influence of the random initial population that has largest effect on the convergence of ECSA. This can be achieved by using the chaotic maps to generate the initial population, since chaotic maps have several advantages. As well as, the opposite-based learning method (OBL) can be used to search about the optimal solution in the opposite direction of the current solution then compare between both of them (i.e., chaotic population and its opposite). This will lead to increase the chance of the algorithm to find the suitable initial population and increase the convergence rate of the algorithm towards the optimal solution.

CRediT authorship contribution statement

Salima Ouadfel: Data curation, Software, Validation, Writing - original draft. **Mohamed Abd Elaziz:** Conceptualization, Software, Validation, Methodology, Software, Writing - original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the China Postdoctoral Science Foundation under Grant No. 2019M652647.

References

Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved Opposition-Based Sine Cosine Algorithm for global optimization. *Expert Systems with Applications, 90*, 484–500.
Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., Zhang, Y., & Mirjalili, S. (2018). Asynchronous accelerating multi-leader salp chains for feature selection. *Applied Soft Computing, 71*, 964–979.
Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications, 116*(1), 47–160.
Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *Soft Computing, 23*(3), 715–734.
Arora, S., Singh, H., Sharma, M., Sharma, S., & Anand, P. (2019). A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access, 7*, 26343–26361.
Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 169*, 1–12.
Bache, K. & Lichman, M. (2013). UCI machine learning repository.
Chattopadhyay, S., Mishra, S., & Goswami, S. (2016). Feature selection using differential evolution with binary mutation scheme. In *2016 International conference on microelectronics, computing and communications (MicroCom). Durgapur, January 23–25* (pp. 1–6). IEEE.
Chen, Y. P., Li, Y., Wang, G., Zheng, Y. F., Xu, Q., Fan, J. H., & Cui, X. T. (2017). A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications, 83*, 1–17.
Chuang, L. Y., Yang, C. H., Wu, K. C., & Yang, C. H. (2011). A hybrid feature selection method for DNA microarray data. *Computers in Biology and Medicine, 41*(4), 228–237.
Dash, R. (2018). An adaptive harmony search approach for gene selection and classification of high dimensional medical data. *Journal of King Saud University - Computer and Information Sciences*.
Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(1–4), 131–156.
De Souza, R. C. T., dos Santos Coelho, L., De Macedo, C. A., & Pierezan, J. (2018). A V-shaped binary crow search algorithm for feature selection. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8). IEEE.
Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18.
Díaz, P., Pérez-Cisneros, M., Cuevas, E., Avalos, O., Gálvez, J., Hinojosa, S., & Zaldivar, D. (2018). An improved crow search algorithm applied to energy problems. *Energies, 11*(3), 571–593.
Djellali, H., Djebbar, A., Zine, N. G., & Azizi, N. (2018). Hybrid artificial bees colony and particle swarm on feature selection. In *Computational intelligence and its applications: 6th IFIP TC 5 international conference, CIIA 2018, Oran, Algeria, May 8–10. 522* (pp. 93–105). Springer.
Dos Santos Coelho, L., Richter, C., Mariani, V. C., & Askarzadeh, A. (2016). Modified crow search approach applied to electromagnetic optimization. In *2016 IEEE conference on electromagnetic field computation (CEFC). USA, Nov 13–16* (pp. 6).
Ekbal, A., & Saha, S. (2015). Joint model for feature selection and parameter optimization coupled with classifier ensemble in chemical mention recognition. *Knowledge-Based Systems, 85*, 37–51.
Elaziz, M. E. A., Ewees, A. A., Oliva, D., Duan, P., & Xiong, S. (2017). A hybrid method of sine cosine algorithm and differential evolution for feature selection. In *2017 International conference on neural information processing, Guangzhou, China, November 14–18* (pp. 145–155). Cham: Springer.
Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381.
Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A. Z., Mirjalili, S., & Fujita, H. (2018). An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems* (154, pp. 43–67). .
Gong, W., & Cai, Z. (2013). Differential evolution with ranking-based mutation operators. *IEEE Transactions on Cybernetics, 43*(6), 2066–2081.
Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2006). *Feature extraction: Foundations and applications (Studies in fuzziness and soft computing)*. New York Inc, Secaucus, NJ, SA: Springer-Verlag.
Hancer, E., Xue, B., Karaboga, D., & Zhang, M. (2015). A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing, 36*, 334–348.
Hassanien, A. E., Rizk-Allah, R. M., & Elhoseny, M. (2018). A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems. *Journal of Ambient Intelligence and Humanized Computing*, 1–25.
Javaid, S., Ali, I., Mushtaq, N., Faiz, Z., Sadiq, H. A., & Javaid, N. (2017). Enhanced differential evolution and crow search algorithm based home energy management in smart grid. In *International conference on broadband and*

wireless computing, communication and applications (BWCCA), Taichung, Taiwan, October 27–29 (pp. 73–86). Cham: Springer.
Javaid, N., Mohsin, S. M., Iqbal, A., Yasmeen, A., & Ali, I. (2018). A hybrid bat-crow search algorithm based home energy management in smart grid. In *Conference on complex, intelligent, and software intensive systems (CISIS). Matsue Japan, July 4–5* (pp. 75–88). Cham: Springer.
Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8*(1), 687–697.
Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing, 147*, 271–279.
Kumar, A., Pakhali, A., Trivedi, I. N., & Jangir, P. (2016). Security constrained nonconvex economic load dispatch problems with prohibited operation zones and ramp rates solve using crow search algorithm. In *AEPEE2016 Morbi, Javier 26* (pp. 28–29).
Lee, W., Stolfo, S. J., & Mok, K. W. (2000). Adaptive intrusion detection: A data mining approach. *AI Review, 14*(6), 533–567.
Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A. Z., & Mirjalili, S. (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems, 145*, 25–45.
Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing, 62*, 441–453.
Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2019). Efficient hybrid nature-inspired binary optimizers for feature selection. *Cognitive Computation*, 1–26.
Mafarja, M., & Sabar, N. R. (2018). Rank based binary particle swarm optimization for feature selection in classification. In *Proceedings of the 2nd international conference on future networks and distributed systems, Amman, Jordan – June 26–27* (pp. 1–19). ACM.
Maheshwari, R., Kumar, M., & Kumar, S. (2016). Optimization of feature selection in face recognition system using differential evolution and genetic algorithm. In *Proceedings of fifth international conference on soft computing for problem solving: 2* (pp. 363–374). Singapore: Springer.
Majhi, S. K., Sahoo, M., & Pradhan, R. (2019). Oppositional Crow Search Algorithm with mutation operator for global optimization and application in designing FOPID controller. *Evolving Systems*, 1–26.
Mirjalili, S. (2016). SCA, a sine cosine algorithm for solving optimization problems. *Knowlede-Based System, 96*, 120–133.
Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191.
Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing, 43*, 117–130.
Nakamura, R. Y., Pereira, L. A., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X. S. (2012). BBA: A binary bat algorithm for feature selection. In *2012 25th SIBGRAPI conference on graphics, patterns and images, Ouro Preto, Brazil, August 22–25* (pp. 291–297). IEEE.
Ng, K. S., & Liu, H. (2000). Customer retention via data mining. *AI Review, 14*(6), 569–590.
Pourpanah, F., Shi, Y., Lim, C. P., Hao, Q., & Tan, C. J. (2019). Feature selection based on brain storm optimization for data classification. *Applied Soft Computing, 80*, 761–775.
Qu, C., & Fu, Y. (2019). Crow search algorithm based on neighborhood search of non-inferior solution set. *IEEE Access, 7*, 52871–52895.
Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing, 74*, 634–642.
Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248.
Rodrigues, D., Yang, X. S., De Souza, A. N., & Papa, J. P. (2015). Binary flower pollination algorithm and its application to feature selection. In Recent advances in swarm intelligence and evolutionary computation (pp. 85-100). Springer, Cham.
Rui, Y., Huang, T. S., & Chang, S. (1999). Image retrieval: current techniques, promising directions and open issues. *Visual Communication and Image Representation, 10*(4), 39–62.
Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2017). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications, 1*, 1–18.
Shaheen, A. M., & El-Sehiemy, R. A. (2017). Optimal allocation of capacitor devices on MV distribution networks using crow search algorithm. *CIRED-Open Access Proceedings Journal, 1*, 2453–2457.
Sharawi, M., Zawbaa, H. M., & Emary, E. (2017). Feature selection approach based on whale optimization algorithm. In *Proceedings of the ninth international conference on advanced computational intelligence, Doha, Qatar, 4–6 February* (pp. 163–168).
Shi, Z., Li, Q., Zhang, S., & Huang, X. (2017). Improved crow search algorithm with inertia weight factor and roulette wheel selection scheme. In *2017 10th international symposium on computational intelligence and design (ISCID). Hangzhou, China, December 9–10* (pp. 205–209). IEEE.
Shunmugapriya, P., & Kanmani, S. (2017). A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and Evolutionary Computation, 36*, 27–36.
Sikdar, U. K., Ekbal, A., Saha, S., Uryupina, O., & Poesio, M. (2015). Differential evolution-based feature selection technique for anaphora resolution. *Soft Computing, 19*(8), 2149–2161.
Sindhu, R., Ngadiran, R., Yacob, Y. M., Zahri, N. A. H., & Hariharan, M. (2017). Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Computing and Applications, 28*(10), 2947–2958.
Singh, N., & Singh, S. B. (2017b). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, 1–15.
Singh, N., & Singh, S. B. (2017a). A novel hybrid GWO-SCA approach for optimization problems. *Engineering Science and Technology, an International Journal, 20*(6), 1586–1601.
Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.
Swets, D. L., & Weng, J. J. (1995). Efficient content-based image retrieval using automatic feature selection. In *IEEE int'l symp. computer vision* (pp. 85–90).
Too, J., Abdullah, A., Mohd Saad, N., Mohd Ali, N., & Tee, W. (2018). A New competitive binary Grey Wolf optimizer to solve the feature selection problem in EMG signals classification. *Computers, 7*(4), 58.
Wang, H., Wu, Z., Rahnamayan, S., Li, C., Zeng, S., & Jiang, D. (2011). Particle swarm optimisation with simple and efficient neighbourhood search strategies. *International Journal of Innovative Computing and Applications, 3*(2), 97–104.
Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.
Xing, E., Jordan, M., & Karp, R. (2001). Feature selection for high dimensional genomic microarray data. In *Proc. 15th int'l conf. machine learning* (pp. 601–608).
Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimization for feature selection in classification: Novel initialization and updating mechanisms. *Applied Soft Computing, Elsevier, 18*, 261–276.
Xue, B., Zhang, M., Browne, W., & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation, 20*(4), 606–626.
Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2019). CCSA: Conscious neighborhood-based crow search algorithm for solving global optimization problems. *Applied Soft Computing, 105583*.
Zawbaa, H. M., Emary, E., Parv, B., & Sharawi, M. (2016). Feature selection approach based on moth-flame optimization algorithm. In *2016 IEEE congress on evolutionary computation (CEC), Vancouver, BC, Canada, July 24–29* (pp. 4612–4617). IEEE.
Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications, 62*, 91–103.