A modified artificial bee colony algorithm for classification optimisation

Selcuk Aslan*

Department of Aeronautical Engineering,
Erciyes University,
Kayseri, Turkey
Email: selcukaslan@erciyes.edu.tr
*Corresponding author

Sibel Arslan

Department of Software Engineering,
Cumhuriyet University,
Sivas, Turkey
Email: sibelarslan@erciyes.edu.tr

Abstract: The promising capabilities, easily implementable and customisable structures of the meta-heuristic algorithms have increased the researchers' attentions to the well-known problems and their new approximations that are suitable to be solved with the meta-heuristics directly. In this study, an attempt was made to solve with an artificial bee colony (ABC)-based technique called classifierABC algorithm, a new approximation that defines the classification problem by using a set of linear equations. The performance of the classifierABC was investigated in detail by using various datasets and assigning different values to the algorithm specific control parameters. The results obtained by the classifierABC algorithm were also compared with the results of the other meta-heuristics including particle swarm optimisation (PSO), differential evaluation (DE), fireworks algorithm (FWA) and different variants of the FWA. Comparative studies showed that the classifierABC solves the new problem approximation more robustly and its solutions determine the classes of instances in sets with high accuracies.

Keywords: meta-heuristics; ABC algorithm; classification optimisation.

Reference to this paper should be made as follows: Aslan, S. and Arslan, S. (2022) 'A modified artificial bee colony algorithm for classification optimisation', *Int. J. Bio-Inspired Computation*, Vol. 20, No. 1, pp.11–22.

Biographical notes: Selcuk Aslan received his MSc and PhD from the Department of Computer Engineering, Erciyes University, Kayseri, Turkey, in 2013 and 2017, respectively. He is currently an Associate Professor with the Department of Aeronautical Engineering, Erciyes University, Kayseri, Turkey. His research interests include meta-heuristics, applications of meta-heuristics in engineering optimisation, parallel and distributed computation.

Sibel Arslan received her MSc and PhD from the Department of Computer Engineering, Erciyes University, Kayseri, Turkey in 2014 and 2019, respectively. She is currently an Assistant Professor with the Department of Software Engineering, Cumhuriyet University, Sivas, Turkey. Her research interests include meta-heuristics, automatic programming techniques and their real world implementations.

1 Introduction

Classification is commonly used to describe a process in which elements of a set of data are tried to be categorised into previously introduced classes or labels. Over the past decades, various artificial intelligence techniques including logistic regression (Guerrero et al., 2021), Naive Bayes (Kotsiantis et al., 2006; Raikar et al., 2020), k-nearest neighbour (Pan et al., 2020), decision tree (Naganandhini and Shanmugavadivu, 2019), random forest (Mohana et al., 2021), artificial neural network (Thurn et al., 2021) and deep learning (Thaiyalnayaki and Joseph, 2021; Shanid and Anitha, 2021) have been used successfully for classifying incoming data of different sources such as social media (Arrigo et al., 2021), network traffic (Wu et al., 2015) and medical measurements (Shrivastava et al., 2019; Zemmal et al., 2021) by utilising from the classified or labelled set of data. In addition to all of these methods, the recent years of computer and information sciences have witnessed to the usage of evolutionary or swarm intelligence-based meta-heuristics on the classification.

Artificial bee colony (ABC) algorithm developed by guiding special foraging habits of real honey bees is one of the most popular meta-heuristics and its variants have improved solving capabilities of some well-known techniques for different problems including classification. Zhang et al. proposed scaled chaotic ABC (SCABC) algorithm and used it for training forward neural network. Experimental studies proved that the SCABC algorithm increases the performance of the forward neural networks on classifying magnetic resonance brain images compared to the forward neural networks trained by back-propagation, magnetic back-propagation, genetic algorithm (GA), elite GA or standard ABC algorithm (Zhang et al., 2011). Prasartvit et al. (2011) selected feasible set of data features with a binary coded ABC algorithm and support vector machine (SVM) was used to classify data with the features chosen by the binary coded ABC. Chandrakala and Sumathi (2012) integrated the ABC algorithm into their image retrieval system to determine fusion weights and improve classification accuracy. Muthuvel et al. (2015) described a three spectrum technique for the classification of ECG signals where ABC and GA train the beat signals in the neural network. Feng et al. (2015) described a modified ABC and GA in order to enhance the hierarchical fuzzy model that determines the labels of the examples more robustly when appropriate membership function is chosen.

Koylu introduced a new online learning algorithm by using the ABC and named this new approach as the online ABC miner. The results of the experiments informed that online ABC miner is a powerful classification model especially for the data streams (Koylu, 2017). Shunmugapriya and Kanmani (2017) proposed a method named as ant colony-artificial bee colony (AC-ABC) to increase both classification accuracy and feature selection capability. Rajaguru and Prabhakar applied Bayesian linear discriminant analysis (BLDA) and hybrid artificial bee colony-particle swarm optimisation (ABC-PSO) for classification of two-phase epilepsy measurements. The combined method presented by Rajaguru and Prabhakar (2017) proved its promising performance in terms of classification accuracy compared to other classifiers. Alshamlan (2018) combined quantitative rule mining and ABC algorithm for defining dynamic quantitative rule-based classification model and used it to classify gene expression profiles in micro array cancer sets. Neagoe and Neghina (2018) presented an ABC-based approach to optimise polynomial discriminant functions and they successfully classified remote sensing images. Barani and Mirhosseini (2018) combined binary ABC algorithm and SVM for defining a new classifier called MABC-SVM. In another study, Rajaguru and Prabhakar (2018) first tried to reduce dimensions of EEG data with the help of local linear embedding and hessian local linear embedding and then categorised processed EEG data with a hybrid ABC-PSO approach.

Xie et al. (2019) introduced an unsupervised band selection method and used ABC algorithm to optimise the combination of selected bands with the guidance of improved subspace decomposition and maximum entropy for hyper-spectral image classification. Zhang et al. (2021) described an attention mechanism to refine the word embeddings and supported it with ABC and SVM classifier for solving the aspect-level sentiment classification problems. Kapila and Bhagat proposed an enhanced ABC algorithm which is called the hybrid fruit fly and ABC (HFFABC) and tested the performance of the new technique for brain tumor segmentation and classification. The experimental studies demonstrated that the HFFABC is capable of obtaining more promising results compared to some of the existing methods (Kapila and Bhagat, 2021). Zorarpaci and Ozel (2021) determined classification rules with an ABC-based technique and showed that their ABC-based technique performs better than other well-known classifiers such as SVM, C4.5, Holte's one rule, PART and RIPPER. Onlooker bee phase of the ABC algorithm was used for hybridisation with the SVM-based technique when selecting features from the SAR images by Rostami and Kaveh (2021) and they proved that proposed feature selection approach increases the number of correctly classified examples. Albkosh et al. (2021) optimised the parameters of gray level co-occurence matrix by using ABC algorithm and increased the texture classification accuracy. Erkan et al. (2022) integrated ABC algorithm as a hyperparameter optimiser for deep convolutional neural network and newly proposed approach was tested for determining correct classes of the plant leaves.

When the literature summary about the implementations of the ABC algorithm on classification is analysed, it is easily seen that ABC algorithm is used in the workflow of a classification technique for further improving its performance by assigning required values to special control parameters, selecting a set of reduced features or it is hybridised with another meta-heuristics or well-known analytical methods. However, a recent description of the classification by Xue et al. (2017, 2018a, 2018b) states that classification is also an optimisation problem that requires solving a set of linear equations and meta-heuristics can be used directly without requiring other techniques. Even though a meta-heuristic algorithm can be used directly for solving the classification problem with the new definition, the existence of dataset being classified and some of its properties such as amount and type of features, number of classes and examples increase the difficulty of classification optimisation and meta-heuristics should be specialised by considering the mentioned situations. In this paper, we first remodelled the employed bee phase of the standard ABC algorithm for improving the limited exploration capabilities. Moreover, onlooker bee phase was changed with a more straightforward technique in which each onlooker forager is sent from hive to the best food source found until current evaluation in order to provide a continuous search within vicinity of it and then a new ABC variant called classifierABC was proposed for solving the classification optimisation problem. The rest of the paper is organised as follows: Details of the new definition of the classification problem was introduced in Section 2.

The phase divided structure of the ABC algorithm was summarised in Section 3 and newly proposed technique for classification optimisation was introduced in Section 4. The results of the experiments and comparative studies were presented in Section 5. Finally, Section 6 was devoted to the conclusion and future works.

2 Definition of classification optimisation

A dataset for which 70% of it contains M different instances or examples each has D features can be represented with an X matrix of size $M \times D$. In addition, labels or classes of M different instances can be showed with Y vector. If a weight vector $W = \{w_1, w_2, ..., w_D\}$ that satisfies the equation (1) can be found, it is understood that label or class of each instance from the X matrix is guessed correctly (Xue et al., 2020a,b; Xue, 2020).

$$X \times W^T = Y.$$

or

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MD} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}. \tag{1}$$

Even though the equation (1) is relatively simple, a solution described with the W vector is found if $R(X) = R(X, Y) < D$ where $R(X)$ is the rank of the matrix X and $R(X, Y)$ is the rank of the matrix generated by concatenation of X and Y (Xue et al., 2020a). However, in most cases, number of instances or examples is greater than the number of features and finding an exact solution for the equation (1) can not be possible all the time (Xue et al., 2020a). By considering the difficulty stemmed from finding an exact solution for the equation (1), Xue et al. (2020a, 2020b) and Xue (2020) decided that the classification decision of an instance can be made by using an approximate solution and some properties of the calculations can be changed as in the equation (2) given below.

$$A \times W^T \approx Y.$$

or

$$\begin{bmatrix} w_1x_{11} + w_2x_{12} + \dots + w_Dx_{1D} \approx y_1 \\ w_1x_{21} + w_2x_{22} + \dots + w_Dx_{2D} \approx y_2 \\ \dots \\ w_1x_{M1} + w_2x_{M2} + \dots + w_Dx_{MD} \approx y_M \end{bmatrix}. \tag{2}$$

When the details of approximation-based definition of the classification problem are analysed, it is seen that the mentioned definition of the classification problem converts it into a continuous numerical optimisation problem with an objective function that will be minimised as in the equation (3) and gives a chance of meta-heuristic algorithms to test their solving capabilities directly on the classification or classification optimisation (Xue et al., 2020a, 2020b; Xue, 2020).

$$f = \sqrt{\sum_{i=1}^{M} \left(\sum_{j=1}^{D} w_j x_{ij} - y_i \right)^2}. \tag{3}$$

Each meta-heuristic algorithm obtains a final solution or a weight vector $W = \{w_1, w_2, ..., w_D\}$ by aiming at the minimisation of the objective function given in equation (3) at the end of run. Even though the objective function values provide important information about the qualities of the solutions, correspondence of the final solution or weight vector should be validated by controlling whether the classes of the examples in the X are correctly determined or not. In order to decide that how the sum of $w_1x_{i1} + w_2x_{i2} + ... + w_Dx_{iD}$ approximates to y_i where i is selected from set $\{1, 2, ..., M\}$ sequentially, a threshold value showed by δ can be defined and used (Xue, 2020; Xue et al., 2022). By guiding the defined δ threshold, classes of the individuals in the X matrix can be determined with the equation (4) (Xue, 2020; Xue et al., 2022). As easily seen from the equation (4), if the result of $w_1x_{i1} + w_2x_{i2} + ... + w_Dx_{iD} - y_i$ is equal or bigger than the $-\delta$ and less than the δ, it is assumed that the label of the x_i instance is determined appropriately.

$$\begin{bmatrix} -\delta \le w_1x_{11} + + \dots + w_Dx_{1D} - y_1 < \delta \\ -\delta \le w_1x_{21} + + \dots + w_Dx_{2D} - y_2 < \delta \\ \dots \\ -\delta \le w_1x_{M1} + + \dots + w_Dx_{MD} - y_M < \delta \end{bmatrix}. \tag{4}$$

The final situation that should be considered for the classification optimisation is about how the upper bound abbreviated as ub and lower bound abbreviated as lb of the elements of vector W are determined. For determining the upper and lower bounds of the elements of vector W, Xue et al. introduced a calculation schema as in the equation (5) (Xue, 2020; Xue et al., 2022). In equation (5), N shows the number of instances in the dataset and σ is a constant that is used to adjust the distance between the upper and lower bounds (Xue, 2020; Xue et al., 2022).

$$(ub, lb) = \pm\sigma \frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} \sum_{j=1}^{D} x_{ij}}. \tag{5}$$

3 ABC algorithm

The honey bees can be classified into three groups called employed, onlooker and scout bees by considering the foraging habits of them. An employed bee is capable of finding food sources, carrying extracted nectar to the hive and informing onlookers about the location and nectar quality of the discovered food sources (Amiri and Dehkordi, 2018). Onlooker bees wait on the hive and select food sources by guiding the provided information. When an onlooker bee decides to select a food source, she becomes an employed bee and contributes foraging operations as an employed bee. The final group of bees generates a small percent of the whole population and tries to discover new food sources without guiding information provided by the employed foragers (Amiri and Dehkordi, 2018). The well-balanced searching routines of the employed, onlooker and scout bees became the source of inspiration for Karaboga and a meta-heuristic called ABC algorithm was proposed. The food sources discovered, consumed and abandoned by the honey bees are matched with the possible solutions of the problem being optimised and nectar quality of a food source is directly related with the appropriateness of the solution. For obtaining optimal or near optimal solutions of the problem, ABC algorithm executes employed, onlooker and scout bee phases sequentially until reaching the determined termination condition is satisfied (Ma et al., 2019; Xiao et al., 2022). Initialisation of the ABC algorithm and details of its bee phases are described in the following subsections.

3.1 Generating initial food sources

In the initialisation stage of the algorithm, all the bees are assumed as scouts and they fly from hive to search and find food sources. Given that there are SN scouts, in other words, there as SN different food sources, and D dimensional problem is tried to be optimised by the ABC algorithm, the equation (6) is used to determine the initial food sources or solutions. While x_{ij} is used on behalf of the j^{th} parameter of the i^{th} solution or food source discovered by the i^{th} scout in the equation (6), $x_j^{\min}$ and $x_j^{\max}$ correspond to the lower and upper bounds of the j^{th} parameter, respectively (Ma et al., 2019; Xiao et al., 2022).

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}). \tag{6}$$

3.2 Employed and onlooker bee phases

When a scout bee finds a food source and starts to memorise it, she becomes an employed forager and tries to discover a new food source within the neighbourhood of the previously memorised one and carries nectar extracted from the related food source to the hive. For modelling the search characteristics of the employed bees within the neighbourhood of the memorised solutions, ABC algorithm utilises from the equation (7) (Ma et al., 2019). In equation (7), v_{ij} represents the randomly determined j^{th} parameter of the v_i food source discovered by the i^{th} employed bee. The remaining parameters of the v_i candidate food source are same with the corresponding parameters of the x_i. While x_{ij} is used to represent the j^{th} parameter of the x_i food source, x_{kj} is matched with the j^{th} parameter of the x_k food source that is selected randomly from the set of food sources containing x_i. However, it should be noted that x_i and x_k are different food sources.

$$v_{ij} = x_{ij} + rand(-1,1)(x_{ij} - x_{kj}). \tag{7}$$

The employed bee related with the x_i food source makes a decision whether the candidate v_i food source is changed with the x_i or not by comparing the qualities or fitness values of them. In ABC algorithm, the fitness value of the v_i food source is determined with the equation (8) for a minimisation problem by using the objective function value of the same food source showed with $obj(v_i)$ (Ma et al., 2019; Xiao et al., 2022). If the $fit(v_i)$ fitness value of the v_i solution is greater than the $fit(x_i)$ fitness value of the x_i, employed bee memorises the v_i solution after setting $trial_i$ counter to zero. Otherwise, considered employed bee still continues to consumption of the x_i food source after incrementing the current value of the $trial_i$ by one.

$$fit(v_i) = \begin{bmatrix} 1 + obj(v_i); & obj(v_i) > 0 \\ 1/(1 + |obj(v_i)|); & obj(v_i) \leqslant 0 \end{bmatrix}. \tag{8}$$

When all of the employed bees complete their foraging operations and turn back to the hive, they give information about the food sources to the onlooker bees with a set of dance figures. In ABC algorithm, number of onlooker bees is taken equal to the number of employed bees. An onlooker bee watches different employed bees and selects a food source for consuming or searching the vicinity of it as is done by an employed forager. The selection of a food source by onlooker bees is directly relational with the fitness value of this food source. In other words, onlooker bees are willing to choose qualified food source or sources. For modelling the mentioned relationship between the selection of a food source and its fitness value or quality, ABC algorithm utilises from the equation (9) and assigns selection probability for each food source (Ma et al., 2019). As easily seen from the equation (9), the selection probability of a food source increases with the fitness value of it. Until sending SN onlooker bees from the hive, ABC algorithm manages selection operations by using the assigned probabilities (Ma et al., 2019).

$$p(x_i) = \frac{fit(x_i)}{\sum_{j=1}^{SN} fit(x_j)}. \tag{9}$$

3.3 Scout bee phase

The performance and solving capabilities of a meta-heuristic algorithm generally depends on a good balance between exploitation and exploration operations. When the mathematical models used by the ABC algorithm for employed and onlooker bees are investigated, it is seen that exploitation-based operations more dominant compared to the exploration-based operations. For providing the mentioned balance between these two main operations, ABC algorithm introduces scout bee phase. In the scout bee phase of the ABC algorithm, the $trial$ counter of each food source is compared to an algorithm specific control parameter called $limit$ calculated with the equation (10) (Agarwal and Mehta, 2019). In equation (10), while SN and D correspond to the number of food sources and dimensions of the problem, a is a positive decimal constant. The $limit$ value of the ABC algorithm is determined in the initialisation stage and protected until the end of execution. The food source that exceeds the $limit$ at most is abandoned and related employed bee with this food source becomes a scout bee for finding an undiscovered food source (Agarwal and Mehta, 2019).

$$limit = \lceil a \times SN \times D \rceil \text{ and } a \in \mathbb{Q}^+. \tag{10}$$

4 Proposed ABC algorithm for classification optimisation

The employed, onlooker, scout bee phases and used mathematical models of the standard ABC algorithm are capable of obtaining promising results for the different types of optimisation problems. However, experimental results reported in various studies about the ABC-based implementations informed us that it suffers from the slow convergence and usually requires more cycles-generations or function evaluations compared to other well-known meta-heuristics especially for the data-dependent optimisation problems (Aslan, 2020). When the details of the classification optimisation problem are investigated, it is clearly seen that existence of the dataset makes the described classification problem data-dependent. In addition, number of instances, number of features and finally number of classes for a given dataset also increase the difficulties of the classification optimisation (Xue et al., 2020b; Xue, 2020; Xue et al., 2022).

By considering the mentioned specialities about the classification optimisation and bottlenecks of the standard ABC algorithm, we tried to modify and customise some stages of algorithm and a new variant was proposed. One of the main modifications made on the proposed ABC compared to the standard implementation of the same algorithm is related with the candidate generation schema of the employed bee phase. In the standard implementation of the ABC algorithm, an employed bee generates its candidate by changing only one parameter of the memorised solution previously and applies greedy selection between the memorised solution and newly generated candidate solution. Determining a random parameter and changing it appropriately for all of the instances in the train set can not be possible. In order to increase the possibility of finding more robust set of parameters that decreases the train error in total even though the new values of the parameters are not suitable for some of the instances, the employed bee phase is modified as given in Algorithm 1. When the details given in Algorithm 1 are investigated, it is easily seen that an employed bee generates its candidate solution by changing all of the elements belonging to the memorised solution represented with a vector of size $1 \times D$.

The idea lying behind the change on the working routines of the employed bees is directly related with the increasing possibility of finding a set of parameters decreasing the train error in total. However, the positive contribution obtained by the changed employed bee phase should be supported with a subtly configured onlooker bee phase for an overall improvement. In the onlooker bee phase of the standard ABC algorithm, onlooker foragers are willing to choose more qualified food sources or solutions introduced by the employed bees. Even though some of the onlooker foragers choose qualified solution or solutions, some of them are still related with poor food sources and they are sent from hive for generating candidates within the neighbourhood of these poor food sources. By considering the mentioned difficulties of the classification optimisation and new modifications on the employed bee phase, it is understood that more stable and continuous search within the neighbourhood of the qualified solution or solutions are required.

Algorithm 1 Modified workflow of the employed bees

//Employed bee phase
for $i \leftarrow 1, ..., SN$ **do**
  if $evalCounter < MFE$ **then**
    for $j \leftarrow 1, ..., D$ **do**
      Generate v_{ij} by using x_{ij} and x_{kj} as in equation (7).
    end for
    if $fit(v_i) > fit(x_i)$ **then**
      Change x_i with v_i
    end if
    $evalCounter \leftarrow evalCounter + 1$
  end if
end for

Algorithm 2 Detailed steps of the classifierABC

Initialisation:
  Assign value to maximum fitness evaluations (MFE) and $limit$
  Initialise *SN* different food sources by using equation (6)
  Set evaluation counter ($evalCounter$) to SN
  $x_b \leftarrow$ get the best food source
Repeat
  //Employed bee phase
  for $i \leftarrow 1, ..., SN$ **do**
    if $evalCounter < MFE$ **then**
      for $j \leftarrow 1, ..., D$ **do**
        Generate v_{ij} by using x_{ij} and x_{kj} as in equation (7)
      end for
      if $fit(v_i) > fit(x_i)$ **then**
        Change x_i with v_i
        if $obj(x_i) < obj(x_b)$ **then**
          Change x_b with x_i
        end if
      end if
      $evalCounter \leftarrow evalCounter + 1$
    end if
  end for
  //Onlooker bee phase
  $numOfOnlookers \leftarrow 0, cr \leftarrow 1$
  for $i \leftarrow 1...SN$ **do**
    $p(x_i) \leftarrow$ calculate the selection probability with equation (9)
  end for
  while $numOfOnlookers \neq SN$ and $evalCounter < MFE$ **do**
    if $p_{cr} > rand(0, 1)$ **then**
      $numOfOnlookers \leftarrow numOfOnlookers + 1$
      Generate v_b around the x_b by using equation (11)
      if $fit(v_b) > fit(x_b)$ **then**
        Change x_b with v_b
      end if

```
            evalCounter ← evalCounter + 1
        end if
        cr = (cr + 1) mod SN
    end while
    //Scout bee phase
    if evalCounter < MFE then
        i ← get index of abandoned source by checking limit values
        Re-initialise x_i food source as in equation (6)
        if obj(x_i) < obj(x_b) then
            Change x_b with x_i
        end if
        evalCounter ← evalCounter + 1
    end if
Until (MFE) is reached.
```

For this purpose, candidate generation schema of the onlooker bees in the standard ABC algorithm is updated slightly by taking into account the advantageous sides of the existing probabilistic schema and guessed requirements of the problem as given in the equation (11). In equation (11), x_b represents the best solution found so far and x_{bj} corresponds to the randomly selected jth parameter of the same solution. Moreover, while x_c is used on behalf of the selected food source by the current onlooker bee and x_{cj} is matched with the j^{th} parameter of the x_c food source or solution, v_b corresponds to the candidate food source generated by the same onlooker bee within the vicinity of x_b and all parameters of the v_b except the j^{th} one are same with the corresponding parameters of the x_b. If the fitness value of the newly determined v_b is better than the fitness value of the x_b, it is changed with the v_b and subsequent onlooker bee starts its operation for generating a new candidate solution within the neighbourhood of the x_b. Generating candidate solutions by guiding the best food source found so far for all of the onlooker bees helps a sensitive search within the neighbourhood of the best solution. For providing a visual representation how the newly proposed employed and onlooker bee phases work, Figure 1 should be viewed.

$$v_{bj} = x_{bj} + \phi(x_{bj} - x_{cj}). \quad (11)$$

The ABC algorithm that will be used for the classification optimisation sends its employed bees by utilising the approach given in Algorithm 1. Moreover, all of the onlooker bees of this ABC variant are directed to the best solution found until the current evaluation as described in the equation (11). For the subsequent sections of this study, the newly proposed ABC algorithm will be called as the classifierABC. In order to accelerate the coding operations related with the classifierABC, a detailed representation of employed, onlooker and scout bee phases is also provided for the possible researchers in Algorithm 2.

5 Experimental studies

The performances of the classifierABC and standard ABC algorithms were investigated by using the datasets taken from the UCI Machine Learning Repository and number of examples, features, classes of these datasets and used σ constant were summarised in Table 1 (Xue et al., 2018a). The value of the number of food sources that is also equal to the number of scout bees or SN in the initialisation of the algorithm was set to 20 and the $limit$ parameter was taken equal to 2,500. Each dataset was splitted into train and test sets. While the 70% of examples selected randomly from the related dataset generate the train set, remaining examples are devoted to the test set (Xue et al., 2018a). For each dataset, classifierABC and ABC algorithms were tested 30 times with random seeds by setting the maximum function evaluations to 100,000 (Xue et al., 2018a). The best solution found at each run and its objective function value were recorded and then summarised in Table 2.

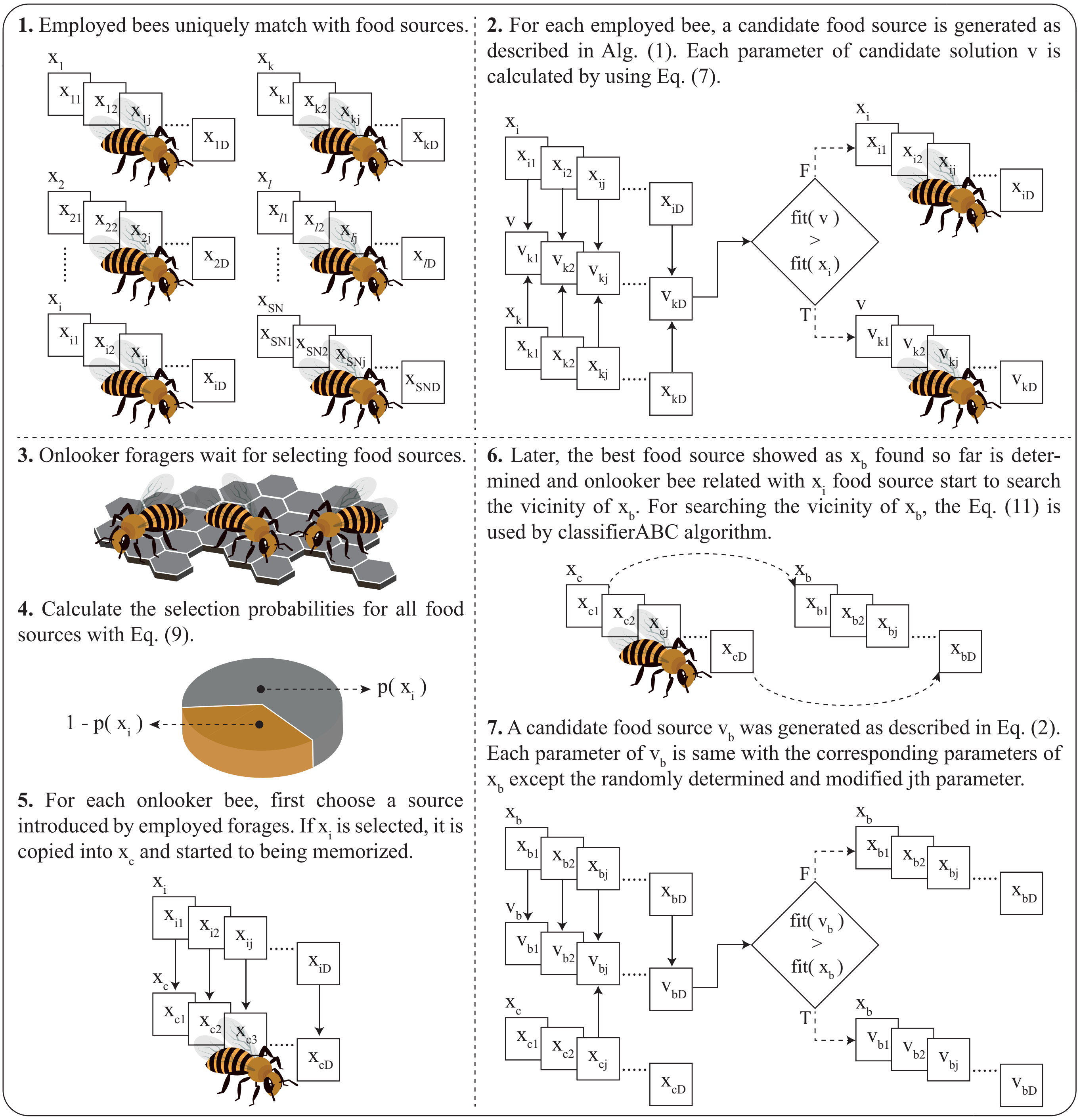
Table 1 Used datasets and their properties

Dataset	*Number of examples/features/classes*	*σ const.*
Biodegradation	1,055/41/2	20
Climate	540/20/2	50
Fertility	100/9/2	30
German	1,000/24/2	50
Ionosphere	351/33/2	50
Iris	150/4/3	50
Spect Heart	267/44/2	30
WBCD	569/30/2	50

Table 2 Results of the ABC-based classifiers

Dataset		*Limit = 2,500*	
		classifierABC	*ABC*
Biodegradation	Mean	*9.792e+00*	1.050e+01
	Best	9.469e+00	9.961e+00
	Std.	1.942e-01	3.173e-01
Climate	Mean	*5.336e+00*	*5.336e+00*
	Best	5.336e+00	5.336e+00
	Std.	3.613e-15	2.507e-06
Fertility	Mean	*2.892e+00*	*2.892e+00*
	Best	2.892e+00	2.892e+00
	Std.	1.355e-15	1.955e-05
German	Mean	*1.110e+01*	1.168e+01
	Best	1.110e+01	1.124e+01
	Std.	9.634e-05	3.383e-01
Ionosphere	Mean	*8.516e+00*	9.898e+00
	Best	8.420e+00	8.870e+00
	Std.	8.601e-02	6.929e-01
Iris	Mean	*2.311e+00*	3.475e+00
	Best	2.310e+00	2.641e+00
	Std.	1.074e-03	1.019e+00
Spect Heart	Mean	*5.039e+00*	5.496e+00
	Best	4.970e+00	5.319e+00
	Std.	4.419e-02	6.655e-02
WBCD	Mean	*5.541e+00*	9.403e+00
	Best	5.475e+00	6.421e+00
	Std.	4.917e-02	3.134e+00

Figure 1 Working routines of the employed and onlooker bees in classifierABC algorithm (see online version for colours)



When the results given in Table 2 are investigated, it is clearly seen that classifierABC algorithm is capable of obtaining better solutions compared to standard ABC for the Biodegradation, German, Ionosphere, Iris, Spect Heart and WBCD datasets. Also, when the Climate and Fertility datasets are considered, the best and mean best objective function values of classifierABC and ABC algorithms are found same for all of the test scenarios. The results given in this table also guide us to roughly generalise that the capabilities of the classifierABC become more apparent compared to ABC with the increasing number of examples or features. The Biodegradation, German, Ionosphere, Spect Heart, Iris and WBCD datasets for which classifierABC outperforms ABC contain more examples, features or both compared to the Climate and Fertility datasets. Each instance in the train set either increases the objective function value tried to be minimised for classification optimisation or the guessed W weight vector perfectly fits with the considered instance and the objective function value does not change. Because of this main reason, minimisation of the objective function for the classification optimisation becomes more difficult with the increasing number of instances. However, in the classifierABC, the employed bee phase searches the solution space in a general manner that focuses on finding a solution or solutions decreasing the objective function value quickly. Moreover, the onlooker bee phase of classifierABC provides a more robust and subtle search by sending all of the onlooker foragers to the neighbourhood of the best solution found until the current evaluation. As an expected result of the mentioned advantageous sides of the classifierABC, it outperforms the ABC algorithm or at least obtains the solutions with the same objective function values.

Another comparison between classifierABC and ABC algorithms was carried out over the convergence performances of them. For a numerical comparison between the convergence performances of the meta-heuristics, Success rate (Sr) and Mean evaluations (Me) are used commonly (Aslan, 2022). If the considered algorithm reaches a solution whose objective function value is equal or less than the previously determined threshold before the end of run, it is assumed that the algorithm is successful for that run and Sr is calculated by dividing number of successful run or runs to total number of runs (Aslan, 2022). If the minimum number of evaluations that is required to reaching the threshold is stored for each successful run and then averaged, the Me is found (Aslan, 2022). In the calculations of the Sr and Me, the threshold is set to 12.0 for the German dataset, 10.0 for the Biodegradation and Ionosphere datasets, 6.0 for the Climate, Spect Heart and WBCD datasets, 3.0 for the Fertility and Iris datasets. When Sr and Me values listed in Table 3 for the classifierABC and ABC algorithms are controlled, it is understood that modified employed and onlooker bee phases contributes to the convergence performance of the classifierABC and its Sr and Me values are found better than the corresponding Sr and Me values of the standard ABC. For the Biodegradation, German, Ionosphere, Iris and WBCD datasets, classifierABC outperforms its competitor when both Sr and Me values are considered. For the Climate, Fertility and Spect Heart datasets, both classifierABC and ABC algorithms reach to the threshold at each of 30 runs and their Sr values are calculated as 100. Even though the Sr values of classifierABC and ABC algorithms are equal to 100, classifierABC still proves its better convergence performance compared to the standard ABC algorithm as easily seen from the Me values.

The objective function values, Sr and Me metrics of the classifierABC algorithm provide important information about its promising performances and capabilities. However, classification optimisation requires another validation in terms of the train and test accuracies calculated as described in the equation (4) by setting δ to 0.5 (Xue et al., 2018a). In the previous test scenarios, the best solution found at each run was used on behalf of the W weight vector and the train and test accuracies were calculated and summarised in Table 4. The results given in Table 4 show that the W weight vector found by the classifierABC determines the classes in the instances of both train and test sets more robustly compared to the standard ABC. The differences between the accuracies belonging to the classifierABC and ABC are more apparent for Biodegradation, German, Ionosphere, Iris, Spect Heart and WBCD datasets and give information about the dominance of the classifierABC over ABC.

Table 3 Sr and Me values of the ABC-based classifiers

Dataset		*Limit = 2,500*	
		classifierABC	*ABC*
Biodegradation	Sr	*86.667*	3.333
	Me	59,253.846	95,280.000
Climate	Sr	*100.000*	*100.000*
	Me	1,340.000	8,144.000
Fertility	Sr	*100.000*	*100.000*
	Me	1,336.000	3,042.667
German	Sr	*100.000*	86.667
	Me	9,356.000	73,173.846
Ionosphere	Sr	*100.000*	66.667
	Me	29,536.000	84,120.000
Iris	Sr	*100.000*	56.667
	Me	12,704.000	37,392.941
Spect Heart	Sr	*100.000*	*100.000*
	Me	484.000	3,901.333
WBCD	Sr	*100.000*	0.000
	Me	11,132.000	-

Table 4 Train-test accuracies of the ABC-based classifiers

Dataset		*Limit = 2,500*			
		classifierABC		*ABC*	
		Train	*Test*	*Train*	*Test*
Biodegradation	Mean	*0.835*	*0.820*	0.804	0.780
	Best	0.855	0.861	0.831	0.845
	Std.	0.009	0.019	0.015	0.023
Climate	Mean	*0.918*	*0.938*	0.913	0.932
	Best	0.918	0.938	0.918	0.938
	Std.	0.000	0.000	0.000	0.002
Fertility	Mean	*0.848*	*0.936*	0.843	0.933
	Best	0.848	0.936	0.843	0.933
	Std.	0.000	0.000	0.000	0.000
German	Mean	*0.757*	*0.747*	0.718	0.725
	Best	0.757	0.747	0.743	0.763
	Std.	0.001	0.000	0.015	0.025
Ionosphere	Mean	*0.765*	*0.710*	0.710	0.657
	Best	0.781	0.743	0.760	0.724
	Std.	0.005	0.021	0.030	0.044
Iris	Mean	*0.963*	*0.956*	0.858	0.821
	Best	0.963	0.956	0.962	0.933
	Std.	0.000	0.000	0.101	0.105
Spect Heart	Mean	*0.812*	*0.760*	0.770	0.753
	Best	0.824	0.800	0.786	0.788
	Std.	0.007	0.018	0.013	0.021
	Mean	*0.941*	*0.918*	0.782	0.786
WBCD	Best	0.945	0.924	0.882	0.900
	Std.	0.003	0.009	0.106	0.107

Table 5 Statistical comparison between ABC-based classifiers

Dataset	*Train*					*Test*				
	Limit = 2,500					*Limit = 2,500*				
	classifierABC vs. ABC					*classifierABC vs. ABC*				
	R+	*R−*	*Z-val.*	*ρ-val.*	*Sign.*	*R+*	*R−*	*Z-val.*	*ρ-val.*	*Sign.*
Biodegradation	3	461	–4.7101	*0.00001*	*classifierABC*	11	453	–4.5456	*0.00001*	*classifierABC*
Climate	-	-	-	-	-	-	-	-	-	-
Fertility	-	-	-	-	-	-	-	-	-	-
German	0	465	–4.7821	*0.00001*	*classifierABC*	25	381	–4.0533	*0.00001*	*classifierABC*
Ionosphere	0	465	–4.7821	*0.00001*	*classifierABC*	23	441	–4.2988	*0.00001*	*classifierABC*
Iris	0	378	–4.5407	*0.00001*	*classifierABC*	0	465	–4.7821	*0.00001*	*classifierABC*
Spect Heart	0	465	–4.7821	*0.00001*	*classifierABC*	110	189	–1.1286	0.25848	-
WBCD	0	465	–4.7821	*0.00001*	*classifierABC*	0	465	–4.7821	*0.00001*	*classifierABC*

Table 6 Comparison between classifierABC and other meta-heuristics for train accuracies

Dataset		*classifierABC*	*ABC*	*DE*	*PSO*	*FWA-CSGS1*	*FWA-CSGS2*	*FWA-CSGS3*	*FWA-CSGS4*	*FWA*	*SaFWA*
Biodegradation	Mean	*0.835*	0.804	0.624	0.483	0.804	0.804	0.803	0.804	0.756	0.805
	Std.	0.009	0.015	0.252	1.100	0.048	0.042	0.044	0.037	0.070	0.048
	Rank	1	3	6	7	3	3	4	3	5	2
Climate	Mean	*0.918*	0.913	0.910	0.773	0.913	0.913	0.914	0.914	0.914	0.915
	Std.	0.000	0.000	0.024	0.767	0.032	0.041	0.039	0.044	0.040	0.037
	Rank	1	4	5	6	4	4	3	3	3	2
Fertility	Mean	0.848	0.843	0.871	0.668	0.880	0.888	0.883	0.889	0.889	*0.891*
	Std.	0.000	0.000	0.000	1.394	0.116	0.088	0.102	0.114	0.115	0.114
	Rank	7	8	6	9	5	3	4	2	2	1
German	Mean	*0.757*	0.718	0.682	0.465	0.730	0.732	0.731	0.729	0.713	0.732
	Std.	0.001	0.015	0.144	0.902	0.060	0.056	0.055	0.051	0.060	0.053
	Rank	1	6	8	9	4	2	3	5	7	2
Ionosphere	Mean	*0.765*	0.710	0.362	0.507	0.753	0.757	0.748	0.760	0.760	0.740
	Std.	0.005	0.030	0.415	1.233	0.117	0.106	0.112	0.107	0.123	0.139
	Rank	1	7	9	8	4	3	5	2	2	6
Iris	Mean	0.963	0.858	0.962	0.700	0.963	0.963	0.961	0.963	0.960	*0.965*
	Std.	0.000	0.000	0.000	1.858	0.058	0.059	0.051	0.052	0.052	0.052
	Rank	2	6	3	7	2	2	4	2	5	1
Spect Heart	Mean	*0.812*	0.770	0.802	0.739	0.799	0.793	0.795	0.793	0.798	0.792
	Std.	0.007	0.013	0.082	0.459	0.080	0.081	0.074	0.093	0.094	0.073
	Rank	1	8	2	9	3	6	5	6	4	7
WBCD	Mean	*0.941*	0.782	0.797	0.714	0.806	0.802	0.805	0.807	0.804	0.803
	Std.	0.003	0.106	0.020	0.711	0.057	0.059	0.057	0.060	0.007	0.073
	Rank	1	9	8	10	3	7	4	2	5	6
Average		1.875	6.375	5.875	8.125	3.500	3.750	4.000	3.125	4.125	3.375
Overall		1	9	8	10	4	5	6	2	7	3

Table 7 Comparison between classifierABC and other meta-heuristics for test accuracies

Dataset		*classifierABC*	*ABC*	*DE*	*PSO*	*FWA-CSGS1*	*FWA-CSGS2*	*FWA-CSGS3*	*FWA-CSGS4*	*FWA*	*SaFWA*
Biodegradation	Mean	*0.820*	0.780	0.629	0.480	0.781	0.793	0.789	0.787	0.742	0.793
	Std.	0.019	0.023	0.259	1.096	0.100	0.110	0.127	0.111	0.129	0.071
	Rank	*1*	6	8	9	5	2	3	4	7	2
Climate	Mean	*0.938*	0.932	0.912	0.804	0.918	0.917	0.914	0.915	0.914	0.912
	Std.	0.000	0.002	0.042	0.818	0.074	0.096	0.091	0.104	0.095	0.088
	Rank	*1*	2	7	8	3	4	6	5	6	7
Fertility	Mean	*0.936*	0.933	0.900	0.664	0.880	0.865	0.866	0.865	0.869	0.855
	Std.	0.000	0.000	0.000	1.550	0.283	0.233	0.258	0.288	0.278	0.258
	Rank	*1*	2	3	9	4	7	6	7	5	8
German	Mean	*0.747*	0.725	0.682	0.467	0.726	0.716	0.720	0.729	0.699	0.725
	Std.	0.000	0.025	0.182	0.920	0.106	0.119	0.114	0.095	0.109	0.088
	Rank	*1*	4	8	9	3	6	5	2	7	4

Table 7 Comparison between classifierABC and other meta-heuristics for test accuracies (continued)

Dataset		*classifierABC*	*ABC*	*DE*	*PSO*	*FWA-CSGS1*	*FWA-CSGS2*	*FWA-CSGS3*	*FWA-CSGS4*	*FWA*	*SaFWA*
Ionosphere	Mean	0.710	0.657	0.359	0.476	0.716	0.695	0.715	0.702	0.709	*0.727*
	Std.	0.021	0.044	0.449	1.198	0.190	0.195	0.224	0.236	0.226	0.209
	Rank	4	8	10	9	2	7	3	6	5	*1*
Iris	Mean	0.956	0.821	*0.978*	0.704	0.954	0.954	0.959	0.959	0.960	0.944
	Std.	0.000	0.105	0.000	1.858	0.111	0.149	0.102	0.129	0.126	0.105
	Rank	4	7	*1*	8	5	5	3	3	2	6
Spect Heart	Mean	0.760	0.753	0.760	0.720	0.781	0.795	0.789	0.795	0.784	*0.797*
	Std.	0.018	0.021	0.163	0.380	0.188	0.191	0.173	0.218	0.221	0.172
	Rank	6	7	6	8	5	2	3	2	4	*1*
WBCD	Mean	*0.918*	0.786	0.794	0.782	0.805	0.807	0.804	0.805	0.807	0.811
	Std.	0.009	0.107	0.048	0.705	0.118	0.140	0.123	0.165	0.165	0.137
	Rank	*1*	7	6	8	4	3	5	4	3	2
Average		2.375	5.375	6.125	8.500	3.875	4.500	4.250	4.125	4.875	3.875
Overall		*1*	7	8	9	2	5	4	3	6	2

Although the differences between the accuracies of the classifierABC and ABC algorithms guide us to decide that the classifierABC is more robust classifier compared to the standard ABC algorithm, this decision should be validated with a statistical test (Ming et al., 2022). For this purpose, Wilcoxon test with the significance level (ρ) equal to 0.05 was used to evaluate classifierABC and ABC algorithms and test results were presented in Table 5. In Table 5, while $R+$ represents the sum of ranks for which the result of the classifierABC is less than the result of ABC, $R-$ represents the sum of ranks for which the result of the classifierABC is higher than the results of the ABC. Also, Z is used on behalf of the standardised test statistics. From the test results given in the mentioned table, it is seen that the differences between the accuracies are capable of obtaining statistical significance in favor of classifierABC for the vast majority of the test cases that include the Biodegradation, German, Ionosphere, Iris, Spect Heart and WBCD datasets. Only for the Climate and Fertility datasets, the solutions found by the classifierABC are not enough to generate statistical difference in favor of the same algorithm.

The experimental studies were completed with the comparison between the classifierABC and other meta-heuristics including differential evaluation (DE) (Xue et al., 2018a; Fan and Yan, 2015), particle swarm optimisation (PSO) (Xue et al., 2018a; Mohiuddin et al., 2016), fireworks algorithm (FWA) (Xue et al., 2018a), FWA-CSGS1 (Xue et al., 2018a), FWA-CSGS2 (Xue et al., 2018a), FWA-CSGS3 (Xue et al., 2018a), FWA-CSGS4 (Xue et al., 2018a) and SaFWA (Xue et al., 2018a). The $limit$ parameter for both classifierABC and ABC was set to 2,500. For guaranteeing that the meta-heuristics obtain their results under the same conditions, the size of the population or number of food sources was taken equal to 20 and maximum evaluation number was set to 100,000 (Xue et al., 2018a). After completing multiple runs, corresponding train and test accuracies were presented in Tables 6 and 7, respectively. When the results given in Tables 6 and 7 are investigated, it is seen that the classifierABC outperforms all of the tested meta-heuristics with the average rank equal to 1.875 for train accuracies and with the average rank equal to 2.375 for the test accuracies. While the Biodegradation, Climate, German and WBCD datasets were processed by the classifierABC with the highest train and test accuracies compared to the other tested meta-heuristics, the Fertility and Iris datasets were processesd by the SaFWA with the highest train accuracies compared to other tested meta-heuristics. In addition these datasets, the classifierABC outperformed its competitors for the Ionosphere and Spect Heart datasets by considering the train accuracy and outperformed its competitors for the Fertility dataset by considering the test accuracy.

6 Conclusions

Because of the versatile and powerful structures, meta-heuristic algorithms have been used successfully for different types of optimisation problems in recent years. As an expected results of these common usage, researchers tried to define approximations directly solvable with the meta-heuristics for well-known problems. In this study, employed and onlooker bee phases of the ABC algorithm were specialised and a new variant also called classifierABC was proposed for solving an approximation to the classification problem. An employed forager of classifierABC tries to generate its candidate after modifying all of the parameters rather than randomly selecting and altering a single one as done by an employed forager of standard ABC algorithm. Moreover, working routines of an onlooker bee in classifierABC were modified in a manner that the neighbourhood of the best food source found until current evaluation is searched with the support of selected food source by the fitness-based probabilistic model.

The performance of the classifierABC was investigated in detail by using different datasets and obtained results were compared with the results of ABC, PSO, DE, FWA and different variants of the FWA such as FWA-CSGS1, FWA-CSGS2, FWA-CSGS3, FWA-CSGS4 and SaFWA. The comparative studies showed that specialised bee phases significantly contribute to the solving capabilities and convergence speed of the classifierABC and it outperforms other tested meta-heuristics by determining labels or classes of the instances with higher accuracies. While the improved exploration capabilities of the employed bee phase give a chance of scanning solution space more robustly for classifierABC, specialised onlooker bee phase of the same algorithm directly sends onlooker foragers in order to examine the vicinity of the qualified solutions. However, it should be noted that mentioned solving specialities of the classifierABC can cause premature convergence especially for the datasets containing relatively small number of features and examples. In future, classifierABC algorithm can be enriched with a method that adjusts the transitions between the working routines of an employed or onlooker bee dynamically by controlling general properties of the dataset being classified. Also, some constants related with the classification optimisation can be changed and their effect on the qualities of the classifications and solving performance of the algorithms can be analysed. Finally, approximation to the classification problem can be modified with a set of nonlinear equations or specialised set of linear or nonlinear equations for each class or label and the performances of the novel meta-heuristics can be analysed in detail.

References

Agarwal, P. and Mehta, S. (2019) 'Abc_de_fp: a novel hybrid algorithm for complex continuous optimisation problems', *International Journal of Bio-Inspired Computation*, Vol. 14, No. 1, pp.46–61.

Albkosh, F.M.A., Mohamed, A.S.S., Elrowayati, A.A. and Awinat, M.M. (2021) 'Features optimization of gray level co-occurrence matrix by artificial bee colony algorithm for texture classification', in *Conference on Engineering Science and Technology*, pp.829–846.

Alshamlan, H.M. (2018) 'DQB: a novel dynamic quantitive classification model using artificial bee colony algorithm with application on gene expression profiles', *Saudi Journal of Biological Sciences*, Vol. 25, No. 5, pp.932–946.

Amiri, E. and Dehkordi, M.N. (2018) 'Dynamic data clustering by combining improved discrete artificial bee colony algorithm with fuzzy logic', *International Journal of Bio-Inspired Computation*, Vol. 12, No. 3, pp.164–172.

Arrigo, E., Liberati, C. and Mariani, P. (2021) 'Social media data and users' preferences: a statistical analysis to support marketing communication', *Big Data Research*, Vol. 24, p.100189.

Aslan, S. (2020) 'An artificial bee colony-guided approach for electro-encephalography signal decomposition-based big data optimization', *International Journal of Information Technology & Decision Making*, Vol. 19, No. 2, pp.561–600.

Aslan, S. (2022) 'An immune plasma algorithm with a modified treatment schema for UCAV path planning', *Engineering Applications of Artificial Intelligence*, Vol. 112, p.104789.

Barani, F. and Mirhosseini, M. (2018) 'Classification of binary problems with svm and a mixed artificial bee colony algorithm', in *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp.1–9.

Chandrakala, D. and Sumathi, S. (2012) 'Application of artificial bee colony optimization algorithm for image classification using color and texture feature similarity fusion', *International Scholarly Research Notices*, Vol. 2012, pp.1–10.

Erkan, U., Toktas, A. and Ustun, D. (2022) 'Hyperparameter optimization of deep cnn classifier for plant species identification using artificial bee colony algorithm', *Journal of Ambient Intelligence and Humanized Computing*, pp.1–12.

Fan, Q. and Yan, X. (2015) 'Differential evolution algorithm with self-adaptive strategy and control parameters for p-xylene oxidation process optimization', *Soft Computing*, Vol. 19, No. 5, pp.1363–1391.

Feng, T-C., Chiang, T-Y. and Li, T-H.S. (2015) 'Enhanced hierarchical fuzzy model using evolutionary GA with modified ABC algorithm for classification problem', in *2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*, pp.40–44.

Guerrero, M.C., Parada, J.S. and Espitia, H.E. (2021) 'EEG signal analysis using classification techniques: logistic regression, artificial neural networks, support vector machines, and convolutional neural networks', *Heliyon*, Vol. 7, No. 6, p.e07258.

Kapila, D. and Bhagat, N. (2021) 'Efficient feature selection technique for brain tumor classification utilizing hybrid fruit fly based ABC and ANN algorithm', *Materials Today: Proceedings*.

Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E. (2006) 'Machine learning: a review of classification and combining techniques', *Artificial Intelligence Review*, Vol. 26, No. 3, pp.159–190.

Koylu, F. (2017) 'Online ABC miner: an online rule learning algorithm based on artificial bee colony algorithm', in *2017 8th International Conference on Information Technology (ICIT)*, pp.653–657.

Ma, L., Wang, X., Shen, H. and Huang, M. (2019) 'A novel artificial bee colony optimiser with dynamic population size for multi-level threshold image segmentation', *International Journal of Bio-Inspired Computation*, Vol. 13, No. 1, pp.32–44.

Ming, F., Gong, W., Wang, L. and Lu, C. (2022) 'A tri-population based co-evolutionary framework for constrained multi-objective optimization problems', *Swarm and Evolutionary Computation*, Vol. 70, p.101055.

Mohana, R.M., Reddy, C.K.K., Anisha, P.R. and Murthy, B.V.R. (2021) 'Random forest algorithms for the classification of tree-based ensemble', *Materials Today: Proceedings*.

Mohiuddin, M.A., Khan, S.A. and Engelbrecht, A.P. (2016) 'Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem', *Applied Intelligence*, Vol. 45, No. 3, pp.598–621.

Muthuvel, K., Suresh, L., Alexander, J. and Veni, S. (2015) 'Spectrum approach based hybrid classifier for classification of ECG signal', in *2015 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, pp.1–6.

Naganandhini, S. and Shanmugavadivu, P. (2019) 'Effective diagnosis of Alzheimer's disease using modified decision tree classifier', *Procedia Computer Science*, Vol. 165, pp.548–555.

Neagoe, V-E. and Neghina, C-E. (2018) 'An artificial bee colony approach for classification of remote sensing imagery', in *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp.1–4.

Pan, Z., Wang, Y. and Pan, Y. (2020) 'A new locally adaptive k-nearest neighbor algorithm based on discrimination class', *Knowledge-Based Systems*, Vol. 204, p.106185.

Prasartvit, T., Kaewkamnerdpong, B. and Achalakul, T. (2011) 'Dimensional reduction based on artificial bee colony for classification problems', in *International Conference on Intelligent Computing*, Springer, pp.168–175.

Raikar, M.M., Meena, S.M., Mulla, M.M., Shetti, N.S. and Karanandi, M. (2020) 'Data traffic classification in software defined networks (SDN) using supervised-learning', *Procedia Computer Science*, Vol. 171, pp.2750–2759.

Rajaguru, H. and Prabhakar, S.K. (2017) 'Bayesian linear discriminant analysis with hybrid ABC-PSO classifier for classifying epilepsy from EEG signals', in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pp.977–980.

Rajaguru, H. and Prabhakar, S.K. (2018) 'Performance analysis of local linear embedding (LLE) and hessian LLE with hybrid ABC-PSO for epilepsy classification from EEG signals', in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp.1084–1088.

Rostami, O. and Kaveh, M. (2021) 'Optimal feature selection for SAR image classification using biogeography-based optimization (BBO), artificial bee colony (ABC) and support vector machine (SVM): a combined approach of optimization and machine learning', *Computational Geosciences*, Vol. 25, No. 3, pp.911–930.

Shanid, M. and Anitha, A. (2021) 'Adaptive optimisation driven deep belief networks for lung cancer detection and severity level classification', *International Journal of Bio-Inspired Computation*, Vol. 18, No. 2, pp.114–121.

Shrivastava, A., Tripathy, A.K. and Dalal, P.K. (2019) 'A SVM-based classification approach for obsessive compulsive disorder by oxidative stress biomarkers', *Journal of Computational Science*, Vol. 36, p.101023.

Shunmugapriya, P. and Kanmani, S. (2017) 'A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC hybrid)', *Swarm and Evolutionary Computation*, Vol. 36, pp.27–36.

Thaiyalnayaki, K. and Joseph, C. (2021) 'Classification of plant disease using SVM and deep learning', *Materials Today: Proceedings*.

Thurn, N.A., Wood, T., Williams, M.R. and Sigman, M.E. (2021) 'Classification of ground-truth fire debris samples using artificial neural networks', *Forensic Chemistry*, Vol. 23, p.100313.

Wu, Q., Chen, J., Ho, S-S., Li, X., Min, H. and Han, C. (2015) 'Multi-label regularized generative model for semi-supervised collective classification in large-scale networks', *Big Data Research*, Vol. 2, No. 4, pp.187–201.

Xiao, S., Wang, W., Wang, H. and Huang, Z. (2022) 'A new multi-objective artificial bee colony algorithm based on reference point and opposition', *International Journal of Bio-Inspired Computation*, Vol. 19, No. 1, pp.18–28.

Xie, F., Li, F., Lei, C., Yang, J. and Zhang, Y. (2019) 'Unsupervised band selection based on artificial bee colony algorithm for hyperspectral image classification', *Applied Soft Computing*, Vol. 75, pp.428–440.

Xue, Y. (2020) 'A classification model based on improved self-adaptive fireworks algorithm', in *Handbook of Research on Fireworks Algorithms and Swarm Intelligence*, pp.148–175, IGI Global, Hershey, Pennsylvania, USA.

Xue, Y., Jiang, J., Xue, B. and Zhang, M. (2017) 'A classification method based on self-adaptive artificial bee colony', in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp.1–8.

Xue, Y., Zhao, B., Ma, T. and Pang, W. (2018a) 'A self-adaptive fireworks algorithm for classification problems', *IEEE Access*, Vol. 6, pp.44406–44416.

Xue, Y., Zhao, B., Ma, T. and Liu, A.X. (2018b) 'An evolutionary classification method based on fireworks algorithm', *International Journal of Bio-Inspired Computation*, Vol. 11, No. 3, pp.149–158.

Xue, Y., Amadou, S.A. and Yan, Z. (2020a) 'Improvement of the firework algorithm for classification problems', *Journal of Cybersecurity*, Vol. 2, No. 4, pp.191–196.

Xue, Y., Zhao, Y. and Slowik, A. (2020b) 'Classification based on brain storm optimization with feature selection', *IEEE Access*, Vol. 9, pp.16582–16590.

Xue, Y., Zhang, Q. and Zhao, Y. (2022) 'An improved brain storm optimization algorithm with new solution generation strategies for classification', *Engineering Applications of Artificial Intelligence*, Vol. 110, p.104677.

Zemmal, N., Azizi, N., Sellami, M., Cheriguene, S. and Ziani, A. (2021) 'A new hybrid system combining active learning and particle swarm optimisation for medical data classification', *International Journal of Bio-Inspired Computation*, Vol. 18, No. 1, pp.59–68.

Zhang, Y-D., Wu, L. and Wang, S. (2011) 'Magnetic resonance brain image classification by an improved artificial bee colony algorithm', *Progress in Electromagnetics Research*, Vol. 116, pp.65–79.

Zhang, M., Palade, V., Wang, Y. and Ji, Z. (2021) 'Attention-based word embeddings using artificial bee colony algorithm for aspect-level sentiment classification', *Information Sciences*, Vol. 545, pp.713–738.

Zorarpaci, E. and Ozel, S.A. (2021) 'Privacy preserving rule-based classifier using modified artificial bee colony algorithm', *Expert Systems with Applications*, Vol. 183, p.115437.