A survey on the studies employing machine learning (ML) for enhancing artificial bee colony (ABC) optimization algorithm

Dervis Karaboga¹,², Bahriye Akay¹* and Nurhan Karaboga³

Abstract: Nature-inspired optimization (NIO) algorithms have gained quite a popularity among the researchers due to their good performance on difficult optimization problems. Recently, machine learning (ML) algorithms dealing with the generation of knowledge automatically from data have been often integrated into NIO algorithms to enhance their performance. One of the widely used popular NIO algorithms is an artificial bee colony (ABC) algorithm mimicking the intelligent foraging behaviour of real honeybees. In order to improve the performance of standard ABC, some hybridization studies of ABC and ML techniques have been performed to introduce more intelligent versions of ABC that can be used for solving the optimization problems arising in ML and other areas. This study presents a survey on the studies combining ABC with ML techniques for enhancing the performance of ABC algorithm and provides a discussion on how ML techniques have been adapted so far and can be employed for improving ABC further. We hope that this study would be very helpful for the researchers dealing with ML and NIO algorithms, particularly ABC.

ABOUT THE AUTHORS

Dervis Karaboga received a Ph.D. degree from Systems Engineering Department of University of Wales, College of Cardiff, UK, in 1994. His general scientific study field is artificial intelligence. Due to his scientific studies, he achieved to be in the list of “highly cited researchers” in the years of 2016–2020.

Bahriye Akay received a Ph.D. degree in Computer Engineering from Erciyes University, Turkey in 2009. Her research areas include artificial intelligence, evolutionary computation, swarm intelligence techniques and software engineering. She has been selected “Highly Cited Researcher” in 2018.

Nurhan Karaboga received a Ph.D. degree in electronics engineering from Erciyes University, Turkey in 1995. Currently, she works as a Professor in the same department. From 1992 to 1994, she also worked as an academic visitor at the University of Wales College of Cardiff, UK. Her current research interests include numerical methods, evolutionary computation algorithms, digital filter design and compressing sensing.

PUBLIC INTEREST STATEMENT

One of the widely used nature-inspired algorithms is an artificial bee colony (ABC) algorithm mimicking the intelligent foraging behaviour of real honeybees. In order to improve the performance of standard ABC, some hybridization studies of ABC and machine learning (ML) techniques have been performed to introduce more intelligent versions of ABC, which can be used for solving the optimization problems arising in ML and other areas. This study presents a survey on the studies combining ABC with ML techniques for enhancing the performance of ABC algorithm and provides a discussion on how ML techniques have been adapted so far and can be employed for improving ABC further.
1. Introduction

Machine learning (ML) and nature-inspired optimization (NIO) (Yang, 2008) are two popular and significant subfields of artificial intelligence (AI). While ML deals with the generation of knowledge automatically from data, NIO exists to cope with the optimization problems met in both AI including ML and other research areas outside AI. In an application of a ML technique for a task, there are several steps to be successfully completed such as collecting sufficient and most meaningful data, extraction and selection of best features, choosing the most convenient ML algorithm, after constructing and training, finding the best model, evaluating the model using the best way, tuning hyperparameters to get the highest performance from the application. In order to solve these problems efficiently, NIO algorithms have been often used recently.

NIO algorithms which are inspired from any natural phenomena (biological or non-biological) are a set of novel optimization methods and they have attracted considerable attention for their simplicity, flexibility and good performance from the researchers working in many different areas (Siddique & Adeli, 2015). Although some algorithms can be considered in more than one group, without loss of generality, they can be mainly divided into two main groups as biology and physics-based algorithms as shown in Figure 1.

a) Biology inspired optimization (BIO) algorithms were inspired from biological phenomena or natural organisms and the algorithms existing in this group can be also classified into two subgroups which are evolutionary algorithms (EAs) inspired by the process of natural evolution...
and swarm intelligence (SI) based algorithms mimicking the intelligent cooperative behaviours of natural creature societies; b) the second group of NIO algorithms consists of physics-based algorithms inspired from physics systems, such as simulated annealing (SA) (Kirkpatrick et al., 1983), gravitational search (GSA) (Rashedi et al., 2009), water drop (WD) (Hosseini, 2007), harmony search (HS) (Geem et al., 2001) and stochastic diffusion search (SDS) (Bishop, 1989) algorithm.

Most of NIO algorithms are population-based and generally speaking, their basic form has a similar and very simple framework, and they usually employ problem independent and probabilistic strategies for their operations. However, the performance of NIO algorithms might be affected by several factors of which some are directly related with the type of operations carried out by the algorithm and the values of control parameters, while some are with the interfacing processes between the problem and the algorithm such as the representation of solutions, defining the objective function(s), and scaling the problem when required. In other words, to get the best performance from an NIO algorithm in an application, these factors must be carefully considered. As expressed above, since the basic form of an NIO algorithm is quite simple and general, its performance might be poor in terms of convergence speed, solution quality, etc., for a particular application. In order to enhance their performance, they have been hybridized with other NIO, meta-heuristic and conventional search algorithms, and hence, several different variants of each NIO algorithm have been developed and presented in the literature. Some versions take their names from the type of optimization problems being addressed by that algorithm, such as floating-point, binary, integer, discrete, combinatorial, multi-objective, dynamic etc. Moreover, since a standard NIO algorithm generally does not use any knowledge extracted from the search experience for guiding the search and employs general probabilistic transition rules, its performance might not be sufficient for all cases. It is very logical that the data related to the different states of the search from the beginning can be collected and then these data can be converted into the knowledge by means of an ML technique. And this induced knowledge can be used for guiding the search more efficiently. For this purpose, different ML techniques have been incorporated into NIO algorithms in different ways to improve their performance and then the new variants have been described. Therefore, another subgroup of the variants of NIO algorithms based on ML algorithms has arisen in the literature.

BIO algorithm group includes older and more popular members of NIO algorithms with respect to the physics-based ones. As expressed above, this group can be divided into two main classes: evolutionary and swarm intelligence-based optimization algorithms. Evolutionary algorithms stimulate the principles of both natural selection and genetic evolution, together. The well-known members of this group are the evolution strategies (ES) (Rechenberg, 1965; Schwefel, 1965), evolutionary programming (EP) (Fogel et al., 1966), genetic algorithm (GA) (Holland, 1975), genetic programming (GP) (Koza, 1990) and differential evolution (DE) (Storn & Price, 1995). Natural swarms such as the societies of bees, ants, fishes, and birds collectively and intelligently behave to solve their particular problems. Also, in order to perform a complex process to demonstrate intelligent behaviours, they do not require any centralized control or a supervision. Swarm intelligence-based algorithms mimic the intelligent cooperative behaviours of these types of natural swarms that arise while solving their problems. Although the first swarm intelligence-based algorithms were introduced only three decades ago, in the 1990s, many new swarm algorithms have been invented and used to solve optimization problems so far, such as ant colony optimization (ACO) (Dorigo et al., 1996), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), artificial bee colony (ABC) (Karaboga, 2005), firefly algorithm (FA) (Yang, 2008), bacterial foraging algorithm (BFA) (Passino, 2002) bees algorithm (BA) (Pham et al., 2006), honey bee mating optimization (HBMO) (Abbass, 2001). Among these, the artificial bee colony is one of the most widely used swarm intelligence algorithms and also, its many new versions have been proposed for solving different type of optimization problems in many different areas (Akay & Karaboga, 2015; Apalak et al., 2014; Karaboga et al., 2014). In several studies, it has been also hybridized with ML techniques to solve the optimization problems arising in the application of ML techniques for a task (Dedeturk & Akay, 2020; Karaboga & Ozturk, 2010; Ozturk & Karaboga, 2011).
Similar to other BIO algorithms, a standard ABC algorithm is quite simple, flexible, probabilistic and problem independent algorithm. However, it does not use any knowledge extracted from the experience gained during the search for further guiding the search more efficiently. Therefore, in order to improve the performance of standard ABC, several hybridization studies of ABC and ML techniques have been performed by means of integrating ML techniques into ABC in convenient ways. This type of hybridization is carried out to introduce more intelligent versions of ABC that can be used for solving the optimization problems existing in both ML and other fields. In this study, our aim is to prepare a survey on the studies combining ABC with ML techniques for enhancing the performance of ABC algorithm and present a discussion how ML techniques have been adapted so far or can be employed for improving ABC in the future. In the study, the publications found from the databases of Web of Science, Scopus and Google by using the keywords “machine learning, statistical learning, artificial bee colony, orthogonal experimental design (OED), opposition-based learning (OBL), clustering analysis, neural network, K-Means, fuzzy C-Means, SVR, SVM, memory” in the title or Keywords of the publications were considered. Totally, 634 papers have been found initially and then 99 publications which are related to the addressed hybridization have been determined after pre-analyzing step. We hope that this survey would be very useful for the readers who are interested in ABC and ML techniques, particularly their combination for different purposes. The rest of the manuscript is organized as the following: The second section of the manuscript describes the standard ABC and then discusses the points where learning techniques can be used. The third section gives the studies which have been carried out on the hybridization of ABC and ML techniques for improving ABC. The section four presents the summary and discussion on the survey, and finally, the conclusion is given in section five.

2. Improving the performance of ABC algorithm with ML techniques

This section firstly introduces the main steps of the basic ABC and secondly explains how learning algorithms can be integrated into these steps for improving its performance.

2.1. Basic ABC

The ABC algorithm proposed by Karaboga (2005) is a successful swarm intelligence algorithm simulating the foraging behaviour of honeybees. In a real honeybee colony, the bees allocated for the foraging are divided into three categories depending on how they search for food sources: employed bees, onlooker bees and scout bees. The employed bees are responsible for loading the nectar of their rich food sources which have been discovered before and saved in their memory. An employed bee shares quality and location information of the food source in her memory with the bees waiting in the hive. The way they share information is dancing. Bees watching the dances of the employed bees are called onlooker bees. The onlooker bees use the dance information gathered from the dances to determine their food sources to visit and exploit their nectar. According to the dance information, higher quality sources are more likely to be selected by the onlooker bees. The other type of forager bees is scout bees that try to discover completely new rich food sources.

The ABC algorithm simulates the food source discovery, selection and exploitation behaviours of foragers of real honeybee swarms. In ABC, each food source is represented by a solution vector of decision variables, and the nectar amount of a solution corresponds to the fitness of the solution. ABC has three phases: employed bee, onlooker bee and scout bee. In the employed bee phase, the vicinity of each food source discovered before is searched by employed bees. The way that the employed bees share information about the quality of sources with the onlookers is mimicked by calculating a probability value for each source, which is proportional to the fitness of the corresponding solution. The onlooker bees perform a roulette-wheel like selection procedure based on the probability values, which means a solution with higher probability value is likely to be selected by an onlooker bee to be exploited. In each exploitation step of a food source (solution), the nectar amount gradually decreases and finally, it is exhausted. This is simulated by assigning a counter to each solution counting the number of exploitations. When this counter reaches a limit value, it means that the associated food source (solution) is exhausted and then it is abandoned by its
employed bee. This employed bee converts into a scout bee and tries to determine a new, unvisited rich food source (solution). Main steps of ABC are given in Alg. 1.

**Algorithm 1. Main steps of ABC Algorithm**

1: Assign values to the control parameters and generate an initial population of D-dimensional food sources (solutions), \( x_i, i = 1, 2, \ldots, SN \), by Equation 1:

\[
x'_i = x_{\min} + \text{rand}[0, 1] \left( x_{\max} - x_{\min} \right)
\]  

(1)

where \( j = 1, \ldots, D \). \( D \) is the number of parameters in the decision vector and \( SN \) is the number of food sources (solutions).

2: Evaluate the fitness \( f_i \) values of food sources (solutions)

3: **REPEAT**

4: Generate neighbour solutions (food sources), \( u_j \), for employed bees using Equation 2 and select the better one between \( x_i \) and \( u_j \):

\[
u_j = x_j + \phi_j(x_j - x_{kj})
\]

(2)

where \( (k \neq i) \in \{1, 2, \ldots, SN\} \) is a randomly chosen neighbour index that is different from \( i \) and \( j \in \{1, 2, \ldots, D\} \) is a randomly chosen dimension index; and \( \phi_j \) is a random number between \([-1, 1]\).

5: Calculate the probability values \( p_i \) for the food sources (the solutions) \( (x_i) \) by Equation 3:

\[
p_i = \frac{\text{fitness}_i}{\sum_{n=1}^{SN} \text{fitness}_n}
\]

(3)

6: Choose food sources \( (x_i) \) by using Equation 3 for onlooker bees and generate neighbour solutions (food sources), \( u_i \), by using Equation 2 and select the better one between \( x_i \) and \( u_i \):

7: Memorize the best solution found so far

8: Determine an exhausted food source (solution), if any, and generate a new solution (food source) for the exhausted one by scout bee randomly using Equation 1

9: **UNTIL**(MaxEvalNumber)

**2.2. Using learning techniques for ABC**

ML techniques can be employed at the different steps of ABC for different purposes to improve its performance. These steps are the initialization of solution population (generation of initial food sources), trial (neighbour) solution production for employed bees, solution selection by onlooker bees, generating trial solution for onlooker bees, abandonment of a solution by its employed bee (scout) and a new unvisited solution generation for the scout bee, fitness evaluation of solutions, adaptation of control parameters and strategies defined for the operators. In the initialization step, instead of using a pure random strategy for the generation of the initial population of solutions, a convenient learning technique benefiting from the data of previously produced
solutions can be employed to sample the search space evenly, and generate higher quality solutions to form an initial population having richer information regarding the solution space.

The operators of a basic ABC are quite simple, flexible, and designed for general purpose, similar to those of other NIO algorithms. Therefore, they are not able to demonstrate the max performance for all type of problems. ML techniques can be used in ABC to learn the structure of the considered problem from the search data and manage the operations depending on this extracted information to direct the search and generate new solutions. To generate a trial solution, neighbour solution(s) are chosen and used in the search equations of employed and onlooker bees in ABC. Most convenient neighbour(s) can be determined by ML algorithms for guiding the search more efficiently, based on the information extracted from the search experience. Another significant point is related to onlooker bees. The solutions of onlookers can be also determined by ML algorithms depending on the collected data. Keeping the diversity in the population of solutions is very significant for the performance of the algorithm, particularly on multi-modal or multi-objective problems. For this aim, the scout bees phase exists in ABC. In basic ABC, a scout bee randomly discovers a solution without using any knowledge. Therefore, an ML technique can be employed to discover richer solutions in an unvisited region of the search space.

There exist many optimization problems to be solved for that designing a correct analytical objective function might be not so easy or even not possible. Moreover, in some of the real-world optimization applications, the evaluation of a solution might take too much time. In other words, placing a possible solution into the considered problem and seeing its effect at the output to calculate the fitness value might require a long time. In these cases, an ML model representing the real objective function can be designed by means of previous solutions and their fitness values obtained to predict the fitness values of the present solutions more quickly. As expected, the performance of an NIO algorithm can be affected by the scale of the considered problem. The scale of large-scale type problems can be decreased by using an ML technique and hence more effective solutions can be generated by the ABC algorithm. Also, an ML algorithm can be employed to restrict the search space and hence the optimization problem is simplified for the optimization algorithm to find better solutions quickly.

Parameter adaptation is a very significant issue to get the best performance from the algorithm for a specific problem. Depending on the evolutionary states, the control parameters of ABC such as colony size, number of onlookers, number of scouts and abandonment criterion (limit) can be adaptively changed. Operator adaptation is another issue to be considered for improving the performance of an NIO optimization algorithm. It means that there exist one or more alternative strategies defined for any operation of the algorithm, and to obtain the best performance from an application, the best set of strategies for the operations must be determined and used in the algorithm according to the information generated from the collected data regarding the optimization process.

3. Studies on using ML techniques for enhancing ABC algorithm

The studies employing learning algorithms for improving the performance of ABC were categorized with respect to the type of learning algorithm or mechanism used such as clustering, reinforcement learning, orthogonal experimental design (OED), opposition-based learning (OBL) and archive and memory. Therefore, the studies were reviewed under six sub-titles in this section.

3.1. Using clustering techniques in ABC

This section describes the publications using K-Means, basic clustering principles and Fuzzy-C-Means for improving the performance of the ABC algorithm. In this group, K-Means is the most frequently used ML algorithm for enhancing ABC. Therefore, the publications are presented under two sub-titles: the studies using K-Means and others.
i) The studies employing K-Means for improving the performance of ABC

K-Means clustering (MacQueen, 1967) aims to partition a data set \( \{x^1, x^2, \ldots, x^N\} \subset \mathbb{R}^n \) into \( C \) clusters by assigning each data point to the cluster with the nearest mean (cluster centres or cluster centroid):

Step 1. Initialize cluster centroids \( \mu_j, j = 1, \ldots, C \)

Step 2. Repeat until convergence is satisfied

\[
\text{Step 2.1} \text{ For each } i = 1, 2, \ldots, N, \quad c^i = \arg \min_j ||x^i - \mu_j^2|| \tag{4}
\]

\[
\text{Step 2.2} \text{ For each } j = 1, 2, \ldots, C, \quad \mu_j = \frac{\sum_{i=1}^{N} 1\{c^i = j\} x^i}{\sum_{i=1}^{N} 1\{c^i = j\}} \tag{5}
\]

Kumar and Sahoo (2017) proposed a two-step ABC algorithm for efficient data clustering, in which the initial solutions are determined using the K-Means algorithm instead of generating randomly. In the onlooker bee phase of ABC algorithm, an enhanced local search operator based the search operator of PSO is employed to explore more promising areas in the search space. Moreover, the abandoned solution position is determined by Hooke and Jeeves-based direct search method. Sun et al. (2015a) presented an improved ABC algorithm based on K-Means clustering, which improves the population diversity using dynamically clustering and enhances the convergence ability by a variation in the information sharing of the employed bees. Sun et al. (2015b) proposed an improved multi-objective ABC algorithm based on K-Means clustering. The convergence rate of the canonical MOABC is enhanced by modifying how information communication is achieved in the employed bees. In order to maintain population diversity, the population is decomposed into clusters using K-Means clustering. Biswas et al. (2012) proposed a hybrid approach which combines K-Means with modified ABC algorithm for locating the clusters in the optima. Sun and Chen (2016) introduced an improved multi-objective ABC algorithm based on K-Means clustering, called CMOABC, in which the population diversity is controlled by decomposition based on K-Means clustering. Since each cluster is evolved solely, after a number of iterations, the population is re-clustered with the aim of information exchange among the clusters. Ibrahim et al. (2020) presented a modified ABC clustering technique for fast and accurate level set segmentation to extract the tumour. In ABC, instead of random initialization, the food source positions are generated using K-Means. The proposed approach determines the cluster centroids and performs level set segmentation to handle topological changes of contours as the brain tumours vary in their form, structure and size. The experiments performed on the BraTs’2017 dataset verify the accuracy of the proposed model. Y. Zhou et al. (2019) combined improved ABC (IABC) algorithm with K-Means clustering for array antenna pattern synthesis. The experimental results reveal that the proposed algorithm enhances the convergence speed and provided a good balance between the local and global search ability in suppressing the sidelobe level and nulling control of the linear array antenna pattern. It yielded faster convergence speed and higher solution accuracy compared to similar algorithms.

ii) The studies employing fuzzy C-means and other clustering principles for enhancing ABC performance

Fuzzy C-Means algorithm (Bezdek, 1981; Dunn, 1973) assigns data points to the clusters by allowing one data point to belong to two or more clusters by a weight. It aims to minimize the objective function given by Equation (6):

\[
J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m ||x^i - c_j^2|| \tag{6}
\]

Step 1. Initialize \( U^0 = [u_{ij}] \)
Step 2. Repeat until convergence is satisfied

Step 2.1 for each $j = 1, 2, \ldots, C$

$$C_j = \sum_{i=1}^{N} u_{mj} x_i$$

(7)

Step 2.2 $u_{ij} = \frac{1}{\sum_{j=1}^{C} \left( \frac{x_{ij}}{x_{ij} + x_{kj}} \right)^{\frac{1}{m}}}$

(8)

A few new variants of ABC have been proposed based on fuzzy C-means (FCM) clustering method. Zhang et al. (2016) presented an advanced ABC algorithm based on FCM clustering method, which targets a balance between the exploitation and exploration. First, the FCM method is utilized to decompose the population into subpopulations so that individuals in the same subpopulation interact only with each other. Additionally, an overlapping area concept has been proposed for the purpose of information sharing among different subpopulations. Krishnamoorthi and Natarajan (2013) proposed an optimization method based on the ABC algorithm for the purpose of clustering. It uses an FCM operator for the scout bees. The experimental results have shown that the proposed approach has provided significant results in terms of the quality of the solution. Zhang et al. (2017) proposed a decentralized ABC algorithm with dynamic multi-populations by means of FCM clustering. During the search, the size of each subpopulation increases in some periods, and the overlapping individuals among different subareas serve to deliver information acting as discovering the search space with the diffusion of solutions. Tran et al. (2020) presented a new hybrid evolutionary approach, called the fuzzy clustering ABC (FABC), to optimize resource assignment and scheduling for non-unit repetitive projects (NRP). In FABC, the fuzzy C-means clustering technique applies several multi-parent crossover operators to utilize population information efficiently and to improve convergence efficiency. Experimental results demonstrate that the proposed method yields the shortest project duration on average and deviation of optimal solution among benchmark algorithms considered herein and those considered previously.

Some researchers have also employed different clustering ideas for improving the performance of the ABC algorithm. Karaboga and Gorkemli (2014) described a quick ABC (qABC) algorithm, a variant of ABC which models the behaviour of onlooker bees more precisely and improves the local search ability of standard ABC. The qABC uses a strategy based on the clustering idea of the employed bees solutions to determine the solutions for onlookers after watching dances. Gorkemli and Karaboga (2019) conducted a study to improve the performance of ABC programming (ABCP). In order to increase the local search ability and achieve higher quality solutions in early cycles, quick ABCP algorithm was developed by using the principle of quick ABC (Karaboga & Gorkemli, 2014), that employs clustering process to determine the solution of an onlooker bee. Görkemli and Al-Dulaimi (2019) applied a novel optimization technique named the quick ABC (qABC) algorithm to the dynamic deployment problem of WSNs. Aslan et al. (2019) described a new exploitation mechanism from the best food source for increasing the convergence performance of the qABC algorithm, where qABC employs a clustering-based strategy to determine the solution of an onlooker bee while protecting the qualities of the final solutions. They analyzed its efficiency on both employed and onlooker bee phases. Moradi et al. (2018) proposed a clustering and memory-based chaotic ABC algorithm, denoted by CMCABC, to solve the dynamic optimization problems. The CMCABC method utilizes an explicit memory to keep the good solutions discovered previously which are not very old. The clustering technique in the proposed method helps to provide diversity. Wen et al. (2020) proposed a multiswarm ABC (MS-ABC) multi-objective optimization algorithm based on clustering calculation and showed that the MS-ABC algorithm exhibits very good performance in medical image registration. Gao et al. (2015) presented an ABC algorithm based on information learning (ILABC). The whole population is clustered into several subpopulations of which size is dynamically adjusted based on the last search experience. Resource management is one of the key concerns in the effective utilization of the mobile grid concept. Vigneswari and Mohamed (2016) proposed a combined ABC and modified heterogeneous earliest finish time (HEFT) based clustering along with a min-min algorithm to generate the initial population and adapted cluster heterogeneous earliest first min-min ABC scheduling algorithm for scheduling the data to computational resource efficiently. Clustering Search (*CS)
combines search algorithms with clustering to discover promising search areas before applying local search operations. Costa and De Oliveira (2013) and Costa & de Oliveira (2015) proposed new approaches based on ABC and *CS, based on the essential properties of promising food sources employed by ABC.

3.2. Using reinforcement in ABC

Reinforcement learning (RL) deals with learning the strategy and strengthening the relations between stimuli, actions, and the occurrence of pleasant events, called rewards (positive reinforcers), or unpleasant events (negative reinforcers) called punishments to maximize the cumulative reward. Assume a reinforcement-learning agent is in a situation $s_t \in S$ at time step $t$, where $S$ is a set of states. Based on its policy, the agent takes an action, $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions for situation $s_t$. When the agent changes its state, $s_t$, to the next state $s_{t+1}$, it receives a reward, $r_{t+1}$, associated with $(s_t, a_t, s_{t+1})$ transition. The aim of reinforcement is that the agent learns a policy, $\pi: AxS \rightarrow [0,1]$, $\pi(a,s) = Pr(a_t = a, s_t = s)$ and maximizes the cumulative reward.

Reinforcement learning method has been also integrated into ABC for enhancing its efficiency in applications. Ma and Zhang (2016) improved the performance of the basic ABC algorithm by adding the reinforcement learning into ABC, in which various updating mechanisms are mapped into an action employed for updating the nectar amount of source position. Zhao and Zhang (2020) presented a novel decomposition-based ABC for MaOP optimization, MoABC/D-LA, to obtain a better generalization ability. The algorithm adapts the searching with a reinforcement learning-based searching strategy in the MoABC/D-LA. Fairee et al. (2019) proposed a combinatorial ABC algorithm based on reinforcement learning update. A reinforcement value is assigned to the solution when a better solution is discovered by an employed bee. Conversely, when a worse solution is produced, negative reinforcement is applied to the solution. Onlooker bees are then recruited to the solutions according to these reinforcement values. Radio frequency identification (RFID) networks planning (RNP) is a challenging task on how to deploy RFID readers under certain constraints. Ma et al. (2019) proposed a specific multi-objective ABC optimizer called H-MOABC, which is based on performance indicators with reinforcement learning and orthogonal Latin squares approach. In the experiments, H-MOABC is used to solve the two scalable real-world RNP instances in the hierarchical decoupling manner. Fairee et al. (2018) proposed a new ABC algorithm by using reinforcement learning in solution updating. According to whether the new solution produced by an employed bee is better or worse, a positive or negative reinforcement applied to the solution to be used by the onlooker bees. The impact of the features responsible for protein complex formation is not uniform. Chowdhury et al. (2013) tried to rank the features required for stable protein-protein complex formation and employed ABC with temporal difference Q learning algorithm to assign weights to the various atomic structure features.

3.3. Using orthogonal experimental design in ABC

Orthogonal experimental design (OED) (Gao et al., 2013; Zhang & Li, 2013) method is a statistic-based method that is employed to extract the knowledge from very limited data, requiring quite low computation cost. OED reduces the number of experiments by systematically planning the experiments (design step) and then the analysis step for statistically analyzing the results. The parameters and the values for the parameters correspond to factors and levels, respectively. Assuming that there are $N$ factors and $Q$ levels per factor, we can construct an $NxQ$ matrix. If $N$ and $Q$ are large, it is too expensive to test all the possible combinations (full factorial design). OED can be used to decide the parameter combinations to be tested for efficient results based on the orthogonal array (OA) and the factor analysis. OA selects the balanced combinations uniformly distributed over the space of configurations.

Orthogonal experimental design (OED) method is a statistic-based method that is employed to extract the knowledge from very limited data, requiring quite low computation cost. As employed
in other EC algorithms, the method has been used in ABC, too. Gao et al. (2013) proposed an improved ABC method where a candidate solution is generated by using a modified search equation to improve the search ability, and furthermore, OED is used to achieve orthogonal learning (OL) strategy to find out more useful information obtained during the search. Zhou et al. (2015b) proposed a parameter extraction approach based on a hybrid ABC (h-ABC) algorithm. In order to tune the control parameters of the h-ABC algorithm, Taguchi’s OED method is used. C. Yang and Guo (2018) presented an orthogonal crossover ABC (OCABC) algorithm based on OED and applied to infer the marine atmospheric duct using the refractivity from clutter technique, and it was also compared with the comprehensive learning PSO and ABC algorithm combined with opposition-based learning and global best search equation. Ren and Wu (2013) proposed a new cooperative coevolution orthogonal ABC (CCOABC) algorithm. Cooperative coevolution frame decomposes the problem into several subcomponents by random grouping and for each subcomponent, the improved ABC algorithm (orthogonal ABC) is employed as the subcomponent optimizer, where an OED method is used to let ABC evolve in a quick and efficient way.

3.4. Using opposition-based learning in ABC

Learning can converge faster when the initial starting point is close to the optimum point while it may require more search budget, or the algorithm may diverge when it is in an opposite location. Therefore, searching the opposite direction and the other directions may help to improve convergence ability (Tizhoosh, 2005). The opposite of a solution vector can be defined by Equation (9):

$$\tilde{x}_i = a_i + b_i - x_i, i = 1, 2, \ldots, n$$  \hspace{1cm} (9)

In each iteration, both the x and \(\tilde{x}\) are evaluated and the learning continues with the better one. The opposite number of the initial guess x will be generated. If the opposite is closer to the solution, then it is taken as a new guess, and the search interval can be recursively halved until the solution and guess are close enough (Figure 2).

Opposition-based learning (OBL) has been more often employed to improve the efficiency of ABC than other ML algorithms. Most of these studies have used OBL technique in the initialization phase of the ABC algorithm. Therefore, the studies integrating OBL into ABC are classified into two subgroups. The first subgroup includes the studies employing OBL in the initialization phase and the second one presents those using it in different phases.

i) The studies using OBL technique in the initialization step of the ABC algorithm

Xiang et al. (2017) introduced a novel scheme for the choice of neighbour solutions in ABC, based on grey relational degrees between a current solution and the neighbour solutions in its vicinity and then, the selected neighbour guides the search process. Moreover, the proposed grey ABC algorithm introduces local search operations employing information of the best individual, an OBL method and a chaotic initialization technique. Y. Wang et al. (2017) proposed a new extreme learning machine (ELM) of which the generalization performance is optimized by a novel ABC. In the ABC, the tent chaotic OBL method is applied to initialize the population, the self-adaptive search strategy is presented in the employed and onlooker bee phases, tent chaotic local search is also implemented for scout bee. The proper allocation of resources is one of the major issues in cloud computing. The objective of the successful resource allocation is minimizing the cost benefits for providers and achieving client happiness. Shameer and Subhajini (2019) used a hybrid OBL and ABC algorithm to find out reliable RA problems in cloud workflow. S. Zhang et al. (2019) proposed a
support vector regression (SVR) model with multi-strategy ABC algorithm (MSABC) for annual electric load forecasting. In the proposed approach, to obtain more diversified initial solutions, tent chaotic opposition-based learning initialization strategy is utilized, and also an improved local neighbourhood search strategy is performed to help the ABC algorithm.

Babaeizadeh and Ahmad (2017) proposed an enhanced constrained ABC algorithm for constrained optimization problems (COPs), where two local search operations are offered for employed and onlooker bee phases, respectively. Besides, both chaotic search method and OBL mechanism are utilized in the initialization to improve the global convergence while producing the initial population. Kong et al. (2016) proposed an effective chaotic ABC approach to global optimization that is applied to nonnegative linear least squares problems. To overcome the insufficiency in the ABC algorithm, OBL initialization method is employed to generate the initial swarm. Furthermore, a new chaotic local search operator is embedded in the algorithm, which can do the local search around the best solution. Yang et al. (2016) presented an improved ABC algorithm called OGABC based on OBL and global best-guided search equation to avoid the slow convergence rate and getting stuck to local optima in the process of inversion of the atmospheric duct. Wang and Zhang (2016) proposed an improved ABC algorithm, where a new search equation for onlooker bees is introduced for enhancing the global convergence, and the initial population is constructed by using a chaotic system and an OBL method. Moreover, to enhance the global convergence rate of the algorithm, a chaotic search is performed around the best solution of the current iteration. W. Mao et al. (2016) developed a new modified ABC algorithm based on the initial population structure, subpopulation groups, step updating, and population elimination. Moreover, based on OBL theory and the new modified algorithms, an improved S-type grouping method is presented, and the roulette wheel selection in the basic ABC algorithm is replaced with the sensitivity-pheromone way. M. Li et al. (2015a) introduced an improved ABC algorithm with a comprehensive search mechanism, that contains three main strategies: the heuristic gaussian search strategy for the employed bees, the best-guided neighbourhood search strategy for onlooker and the self-adaptive population perturbation strategy. In addition, to improve the quality of the initial population, the chaotic OBL method is used for the initialization process. Bhandari et al. (2015) presented a modified ABC algorithm for satellite image segmentation using a different objective function to find the optimal multilevel thresholds. In MABC algorithm, solution search equation is improved by exploiting the best solution of the previous iteration to enhance exploitation, and also when generating initial population, both chaotic system and OBL method are
employed to improve global convergence. F.-J. Kuang et al. (2015) proposed a hybridization algorithm of tent chaos ABC and PSO, where an initialization strategy based on tent chaotic OBL is applied. T.-L. Li et al. (2015b) presented an improved algorithm called DCABC in that the OBL method is employed when producing the initial population and the divide-and-conquer strategy is adopted to greed update food resources. After employed bees releasing updated food source information, onlookers choose optimal resource based on the divide-and-conquer strategy.

Luo and Wang (2014) proposed an improved ABC algorithm for the accurate evaluation of minimum zone axis straightness error from a set of coordinate measurement data points. In the proposed algorithm, the OBL method was used in the initialization and scout bee phases. F. Kuang et al. (2014) proposed anew self-adaptive chaotic ABC algorithm based on Tent map (STOC-ABC) to improve the global convergence and increase population diversity. In the STOC-ABC, Tent chaotic OBL initialization method is introduced to diversify the initial individuals and produce good initial solutions. G. Li et al. (2014) presented an optimization technique based on ABC, where OBL is used in the population initialization, the greedy selection is excluded and the way that an employed bee transforms into a scout is modified. Lai and Qu (2012) proposed a modified strategy of initialization for the standard ABC, which utilizing the logistic map and OBL to generate the initial population as well as the scout bee position. In addition, the employed bee search equation is modified by adding weight coefficients for the purpose of increasing the convergence speed. W. Gao et al. (2012) presented an improved ABC algorithm where, to enhance the global convergence, the OBL method is employed to produce the initial population. El-Abd (2012) enhanced the performance of ABC by introducing the concept of generalized OBL. This concept is introduced through the initialization step and through generation jumping, and the proposed GOBL-based ABC is compared to the performance of ABC and opposition-based ABC (OABC) using the CEC05 benchmarks library. Gao et al. (2011) proposed an improved ABC by modifying its search strategy to obtain a well balance in the exploration and exploitation and to achieve satisfactory optimization performances. Moreover, to intensify the global convergence, both OBL method and chaotic maps are employed when producing the initial population. Gao et al. (2012) proposed a modified ABC algorithm (ABC/best) where each bee searches only around the best solution of the previous iteration, and also to enhance the global convergence, both chaotic system and OBL method are employed when producing the initial population and scout bees. Gao and Liu (2011) presented an improved ABC algorithm for global optimization, and to enhance the global convergence speed, when producing the initial population, both the chaotic systems and the OBL method are employed. El-Abd (2011) enhanced the performance of the ABC algorithm for function optimization by introducing the concept of OBL. This concept is introduced through the initialization step and through generation jumping. Bao and Zeng (2011) proposed a novel bi-group differential ABC algorithm where an initialization strategy based on the OBL is applied to diversify the initial individuals in the search space. All individuals are randomly divided into two populations and their evolutions are simultaneously performed with different optimization strategies, also the interactive learning strategy is introduced to accelerate the convergence speed. Kalaikumar and Baburaj (2020) presented fuzzy-based cross-layer mechanism for the management of congestion using oppositional ABC (FCOABC) protocol for performing inter-cluster multi-hop routing from CHs to master station; and therefore, an energy efficient and reliable data transfer is achieved to the master station. In Sharma and Abraham (2020), the food locations in basic ABC are improved using OBL concept and further enhanced by integrating greediness in searching behaviour. The modifications aim to maintain population diversity and to improve exploitation. The proposed approach is verified on seven mechanical engineering design problems. Liu et al. (2019) proposed an enhanced exploitation ABC algorithm, denoted by EeABC. A generalized OBL strategy (GOBL) is used to generate the initial population to obtain an evenly distributed population. Additionally, two new search operators inspired by DE are proposed. Huang et al. (2019) presented an enhanced hybridized ABC (EHABC) algorithm for optimization problems. To improve the accuracy performance of ABC, the OBL method is employed to produce the initial population.
ii) The studies using OBL technique in other steps of the ABC algorithm

Guo et al. (2019) proposed a chaotic ABC with elite OBL strategy (CEOABC). During the search process, CEOABC utilizes the chaotic local search to improve the exploitation ability. Moreover, the elite OBL strategy is utilized to get profit from the potential information of the exhausted solution. Xiang et al. (2019) proposed a multi-strategy ABC (ABCVNS) based on the variable neighbourhood search method. At the first stage, a pool of search strategy candidates that consists of two search strategies is used in the employed bee and onlooker bee phases. Then, another pool of strategies which consists of the original random search strategy and the OBL method are employed to boost the balance in exploration and exploitation abilities in the scout bee phase. Sharma and Gupta (2018) improved the foraging process of two phases of ABC by incorporating OBL concept, which enhances the acceleration and exploitation capability of ABC and then O-ABC is applied for intrusion detection. Dhaliwal and Dhillon (2016) incorporated the OBL strategy into ABC and hence, introduced a modified version called oppositional ABC algorithm. The developed algorithm was then used for optimal and stable digital IIR filter design. Xiang et al. (2018) proposed a novel ABC (CosABC) based on the cosine similarity, which is employed to choose a better neighbour individual to further improve the convergence rate of ABC, hence a new solution search equation is proposed to minimize the weakness of undirected search of ABC. In the onlooker bees’ phase, ABC/rand/1 is employed to improve the exploitation ability while an OBL technique is also used to balance the exploitation of ABC/rand/1. Zhou et al. (2015) introduced a Gaussian bare-bones ABC where a new search equation is introduced based on information of the global best solution and also the generalized OBL strategy is used to generate new solutions for scout bees, which helps to gather more useful information for guiding search. Zhao et al. (2016) proposed a novel method, called OBL-based ABC to improve the performance of ABC. They introduced four common kinds of opposition-based models into their new approach, where the new generated solution and opposition solution are used to enlarge the searching area and the new location of the scout is decided by the location of the employed bee to enhance local exploitation of the scout. Worasucheep (2015) introduced an opposition-based hybrid of ABC and DE algorithms for solving continuous problems. The proposed algorithm, called OABCDE, adopts mutation operation of DE and a crossover-like mechanism to improve the convergence ability of ABC without introducing new parameters. The OBL is periodically repeated to avoid getting trapped in local optima.

Zhou et al. (2015a) introduced a neighbourhood search mechanism to enhance the solution search equation of the ABC algorithm. Moreover, in order to preserve search experience for scout bees, the generalized OBL strategy was utilized to generate opposite solutions of the discarded food sources, which helps enhance the search efficiency. X.-Y. Zhou et al. (2015b) proposed an improved ABC variant by employing the generalized OBL (GOBL) strategy which helps to discover much more promising search regions and converge to the global optimum. Zhao et al. (2015) presented a novel ABC, called ABC using OBL, which produces opposite solution by the employed and onlooker bees, and chooses the better solution using greedy selection strategy to enlarge the search areas. Wang (2015) proposed a new search strategy for the employed bees phase of ABC by adopting generalized OBL method as a search operator and an improved solution search equation by exploiting the advantages of the local best solution at the onlookers phase. Both operations can balance the exploration and the exploitation of the proposed algorithm. Song et al. (2015) developed an emergency resource scheduling model with two layers of affected points to minimize total cost and reaction times, that is based on an improved ABC. The improved ABC algorithm uses opposition-based and comprehensive learning concepts to obtain Pareto-optimal solution sets and then analyzes the number of solutions for a Pareto front and uniformity measure. Sharma and Pant (2013) suggested some modifications in the basic ABC, called Intermediate ABC (I-ABC) and I-ABC greedy. In I-ABC, the new food sources are produced by using the intermediate positions between the uniformly generated random numbers, and random numbers generated by OBL. In I-ABC greedy, the search is guided towards the best solution in the population. To balance the diversity and convergence capability of the ABC, Sharma et al. (2013) proposed a local search using
Lévy flight random walk and integrated with OBL strategy. The proposed algorithm is named as opposition-based Lévy flight ABC.

Yang and Huang (2012) presented a new ABC optimization algorithm to solve function optimization problems. The proposed approach introduces OBL concept and dynamic cauchy mutation into the standard ABC algorithm. Bi and Wang (2011a) proposed an improved ABC algorithm, where a new crossover strategy is designed to make the group close to the optimal solution as soon as possible. Also, a mutation strategy based on OBL is proposed to replace the scouts’ behaviour. Sharma and Pant (2011) suggested some modifications in the basic ABC to boost its performance. In the proposed algorithm, called intermediate ABC, the potential food sources are produced by using the intermediate positions between the uniformly generated random numbers and random numbers generated by OBL. Bi and Wang (2011b) presented an improved ABC algorithm, called fast mutation ABC algorithm (FMABC). The onlooker bee phase utilizes the pheromone and the sensitivity model to be substituted with a roulette wheel selection model in the basic ABC. A mutation strategy based on OBL was also presented to be used in the scout bee phase. Yang and Dong (2016) proposed an OBL adaptive quick ABC algorithm (OAQABC). The opposition-based learning was integrated into ABC to improve the employed bee phase. The experimental results show that OAQABC has better performance than basic ABC, quick ABC, CS, and PSO. Shao et al. (2020) enhanced the ABC algorithm using refraction principle (EABC-RP). In order to intensify its exploration, the unified OBL (UOBL) based on refraction principle is adopted to produce refraction solutions for employed bees, which provides diversity in the population and guides the search towards the global optimal solution. Besides, in the scout bee phase, the UOBL based on refraction principle is utilized to jump out of the local minima.

3.5. Using archive and memory in ABC

In the literature, there exist some studies employing archive and memory-based ABC to solve the problems. Multi-objective evolutionary algorithms (MOEAs) have been a most suitable approach for solving multi-objective software module clustering problems (M-SMCP). Amarjeet (2018a) proposed a two-archive-based ABC (TA-ABC) algorithm for M-SMCPs containing more than three objective functions. The clustering solution obtained by TA-ABC is compared against genetic-based two-archive algorithm (TAA) and non-dominated sorting genetic algorithm II (NSGA-II) over practical problems. Amarjeet (2018b) proposed a many-objective ABC (MaABC) algorithm to solve many-objective software module clustering problems (SMCP). They modified the basic ABC by using a quality indicator, Lp-norm-based distances, and two external archives concepts. To verify the efficiency of the proposed approach, a comprehensive comparative study is performed with state-of-the-art many-objective optimization algorithms, including Two-Arch2, NSGA-III, MOEA/D, and IBEA over seven SMCPs. Amarjeet and Chhabra (2018d) proposed a fuzzy-pareto dominance driven ABC (FP-ABC) to solve the many-objective software module clustering problems (MaSMCPs) effectively and efficiently, where fuzzy-Pareto dominance and two external archive concepts have been incorporated with the ABC algorithm. The fuzzy-pareto dominance enhances candidate solution selection and two external archives concept aims to gain a balance between the convergence and diversity.

In order to improve the search performance, some ABC variants employ an explicit memory storing the data related to previous states of the search. X. Li et al. (2019) studied on a flexible time-of-use (FTOU) tariff model to optimize the electricity prices and their allocations to different time periods simultaneously, constrained by the dynamic demand of customers. They used a mixed ABC (mABC) approach to manage the continuous prices and discrete allocations simultaneously, embedded with a transferred memory scheme (TMS) to obtain the flexible and smooth tariff design with dynamic demand. Moradi et al. (2018) presented a clustering and memory-based chaotic ABC algorithm, called CMCABC, for solving the dynamic optimization problems. CMCABC employs an explicit memory to keep the previous good solutions which are not very old. Using clustering technique in the proposed method can provide good diversity in the problem environment. Zabihi and Nasiri (2018) proposed a new version of ABC algorithm called history-driven ABC
(Hd-ABC) to enhance the performance by integrating a memory mechanism provides the fitness landscape to be approximated before actual fitness evaluation. Hd-ABC employs a binary space partitioning (BSP) tree to save useful information of the evaluated solutions. Fan et al. (2018) proposed a new ABC variant named hybrid ABC (HABC) algorithm. In HABC, inspired by real honeybees, a memory mechanism is introduced to remember the previous successful experiences and further direct the next foraging behaviour. The proposed memory mechanism aims to improve global search efficiency. Du et al. (2017) described an improved ABC with memory where the candidates converge to the attractors consuming only one function evaluation, and the candidate will replace the current solution if the former is better than the latter. Otherwise, the memory will be deleted directly. M. Mao et al. (2017) presented a modified ABC algorithm (ABCEM) which employs an adaptive search equation for the employed bees and extended memory keeping the employed and onlooker bees’ historical information. Moreover, the extended memory is incorporated with two solution search equations in the employed bees and the onlookers to improve the quality of food sources. Cross-docking consists of interrelated operations which require proper synchronization. Only a few attempts have addressed the vehicle routing problem in this context. Mao and Duan (2016) presented a modified ABC algorithm using self-adaptive extended memory (ABCSEM), where employed bees’ historical information including personal best, global best solutions are stored in the extended memory to improve the exploitation capability. Yin and Chuang (2016) developed an adaptive memory ABC (AMABC) algorithm to tackle this problem. Compared to a tabu search proposed in the literature, the AMABC algorithm can reach higher fuel efficiency by managing the loading along the route and yield less cost and CO₂ intensities. Li and Yang (2016) described a new ABC variant named ABC with memory algorithm (ABCM) which simulates memorizing bees’ previous successful experiences of foraging behaviour. The memory mechanism is applied to guide the further foraging of the artificial bees. Nakano et al. (2015) proposed an improved ABC algorithm (IABC) and then included memory and a detection scheme for dynamic changes to develop the searching performances for dynamic optimization problems (DOPs). Parvin et al. (2018) implemented an ABC approach enhanced with an explicit memory to store the past best solutions and a population clustering scheme to maintain diversity in the population, which will help speed-up the convergence of the algorithm for solving dynamic optimization problems.

3.6. Others
Moreover, there are also a few studies combining ABC and ML algorithms such as case-based reasoning, comprehensive learning. G. Wang et al. (2018a) studied on anti-learning to increase the initial population of ABC and avoid the algorithm falling into local optimum. The global optimal value is introduced into the neighbourhood search formula to accelerate the global convergence speed and also case-based reasoning technology is employed to adjust the search direction in real time and strengthen the dynamic optimization ability. Ren et al. (2019) implemented an algorithm of reliable data collection for mobile Sink based on an improved ABC optimization which performs reverse learning in the initialization and adopts the search equation inspired by DE algorithm. It avoids the premature convergence and poor ability to search in late evolution, which satisfies the conditions of the shortest time consumption and the shortest path length of the mobile Sink.; J. Wang et al. (2018b) proposed an improved ABC algorithm using heterogeneous comprehensive learning (HCLIABC) and double population in which the multiplicative weight update method is carried out to update the selection probability. In the algorithm, the whole population is decomposed into exploration-subpopulation and exploitation-subpopulation. HCL strategy produces the exemplars for two subpopulations and OBL improves the quality of the first population.

4. Summary and discussion
In this paper, we reviewed the studies regarding ML algorithms used in several phases of the ABC algorithm. The total number of publications considered in the survey is 99. Firstly, these publications have been categorized depending on the kind of learning algorithm or mechanism employed in the study. Fifty publications, almost half of the total studies, employ OBL algorithm in ABC. Approximately one quarter, 21 studies, use a clustering process for improving ABC such as K-Means, fuzzy C-means. Five studies benefit from OED, 6 from reinforcement learning, 14 from a memory mechanism and 3
from other learning algorithms, respectively. The distribution of the publications with respect to the learning techniques or mechanisms are presented in Figure 3. From these results, it can be stated that, although there are many different learning algorithms in the literature that have a potential for improving ABC, most of them except OBL have not been integrated into it. Very limited numbers of them have found the use in enhancing the performance of ABC.

Publication numbers with respect to years are presented in Figure 4. It is seen that the first publications were introduced into the literature in 2010 and the publication number was 7 in that year. During the recent five years, it has varied around 15. In summary format, Table 1 presents all studies with respect to the kind of learning algorithm employed. Of course, it is important to know what type of ML algorithm is used for ABC in a study. Moreover, it might be more insightful for the readers to understand in which phases of ABC the ML algorithms are integrated. ML techniques have been used at the different steps of ABC for different purposes to improve its performance. These steps are the initialization of solution population (generation of initial food sources), trial (neighbour) solution production for employed bees, solution selection by onlooker bees, generating trial solution for onlooker bees, abandonment of a solution by its employed bee and a new unvisited solution generation for the scout bee. Most of the studies considered in this review are related to the use of ML algorithm in the initialization, employed, onlooker and scout bee phases. Table 2 demonstrates the phases of the ABC algorithm where the ML algorithms are employed by the studies. From the table, it is seen that OBL technique is heavily used in initialization, employed, onlooker and scout bee phases. For population diversity control, clustering and memory-based techniques have been used.

Instead of using pure random initialization for ABC, a learning technique can be employed to generate an initial population with richer information. For this purpose, OED and ML techniques such as OBL, Neural Network (NN) and interpolation can be used for determining the initial positions of solutions in more promising search regions. So far, only OBL has been used for this purpose. OBL technique aims to form a better initial population by means of produced two opposite population. Interpolation tries to find a better initial solution than previously random generated solutions by applying the interpolating process on these solutions. These are quite simple techniques and can be easily employed in ABC, and the contribution of their use to the
efficiency in the initialization phase can be investigated in detail. Particularly, in the dynamic optimization application of ABC, ML techniques such as cased-based reasoning can be used to extract information from the experience and reinitialize the population or insert new solutions into the population when a change occurred with the problem.

As stated before, in the employed and onlooker phases of ABC, the probabilistic operations are carried out. The generation of trial solutions in both phases, and the food source selection mechanism for onlookers are the main operations. For these operations, new strategies using learning mechanisms can be adapted. As known, the third phase of ABC is the scout bee phase. A scout bee randomly discovers a solution without using any knowledge. Also, a suitable ML technique can be employed to discover a richer and unvisited solution in a promising region. Just a few studies on using learning principles in these three phases have been introduced into the literature, so far. An ML model can be designed by means of the previous solutions and their fitness values and then this model can be used to predict the fitness values of the new solutions more quickly. For this aim, the widely used algorithms are ANN, statistical methods, regression and interpolation techniques. All regression or prediction ML models can be employed for this purpose. Also, rather than evaluating all solutions in the population by using real objective functions, only some of them can be evaluated based on the objective functions. For example, the population can be divided into subclusters and only the representative of each cluster can be evaluated by the real objective function and the fitness values of others can be predicted or determined by a simpler convenient model. As expected, the performance of ABC can be affected by the scale of the problem. By restricting the search space using an ML algorithm, the problem can be simplified for the ABC algorithm. For this purpose, different ML algorithms such as Principal Component Analysis (PCA), Cluster Analysis (CA) or NN can be employed.

For parameter adaptation, usually simple statistical methods are used to evaluate the collected related data and then they adaptively vary the parameters of the algorithm. CA or rule-based systems can be employed for this purpose. Depending on the evolutionary states, the control parameters of ABC, colony size, number of onlookers, number of scouts and limit value can be adaptively changed. Operator adaptation is another significant issue to be considered for improving the performance of an NIO algorithm. In the case of ABC, various strategies have been defined for the initialization of population, trial (neighbour) solution generation, distribution of onlookers, production of scouts. During the search, the best combination of the strategies can be determined for the phases to maximize the overall performance of ABC.
Consequently, so far, very limited kinds of statistical or machine learning algorithms have been hybridized with ABC for improving its efficiency. Also, most of the studies introduced proposed to use learning mechanisms, particularly OBL, in the initialization, employed, onlooker and scout bee phases. The significant drawback of ABC, similar to other NIO algorithms, is that it has a high computation cost. In order to reduce the evaluation number of solutions, convenient ML techniques can be employed. As understood, there exist several problems which are related to ABC itself and its application to an optimization problem and they can be effectively solved by using a suitable ML algorithm.

### 5. Conclusion

This study presented a survey on the studies using ML algorithms for improving the ABC algorithm. Performing a search in Scopus, Google and WoS databases based on widely used keywords related to ML algorithms and ABC, 99 publications were selected. The publications considered in this study demonstrate that the use of ML for the problems occurring in ABC is an active research area. From the evaluation of these publications, it was seen that ML algorithms have been incorporated into ABC in different phases for enhancing the search behaviour of the algorithm. Most of the studies used OBL and clustering processes such as K-Means and Fuzzy C-Means for initialization, employed, onlooker and scout bee phases. There is no study regarding the parameter and operator adaptation of the ABC algorithm. Also, we have not met any study

| Table 1. All studies with respect to the kind of learning algorithm employed |
|---------------------------------|---------------------------------|
| **ML Technique in ABC**         | **Publications**                 |
| **Clustering Techniques**       | Karaboga and Gorkemli (2014),  |
|                                 | Gorkemli and Karaboga (2019),    |
|                                 | Görkemli and Al-Dulaimi (2019), |
|                                 | Aslan et al. (2019), Moradi et   |
|                                 | al. (2018), Wen et al. (2020),   |
|                                 | Gao et al. (2015), Vigneswari     |
|                                 | and Mohamed (2016), Costa and    |
|                                 | De Oliveira (2013; 2015),        |
|                                 | Kumar and Sahoo (2017), Sun et   |
|                                 | al. (2015a), Sun et al. (2015b),|
|                                 | Biswas et al. (2012), Sun and    |
|                                 | Chen (2016), Ibrahim et al. (2020)| |
|                                 | Y. Zhou et al. (2019), Zhang et   |
|                                 | al. (2016), Krishnamoorthi and   |
|                                 | Natarajan (2013), Zhang et al.   |
|                                 | (2017), Tran et al. (2020)       |
| **Reinforcement**               | Ma and Zhang (2016), Zhao and    |
|                                 | Zhang (2020), Fairee et al. (2019)| |
|                                 | Ma et al. (2019), Fairee et al.  |
|                                 | (2018)                           |
| **Orthogonal Experimental Design**| Gao et al. (2013), X.-Y. Zhou     |
|                                 | et al. (2015b), Akkan et al. (2019)| |
|                                 | C. Yang and Guo (2018), Ren and  |
|                                 | Wu (2013)                        |
| **Opposition-based Learning**   | Xiang et al. (2017), Y. Wang et  |
|                                 | al. (2017), Shameer and Subhajini |
|                                 | (2019), Gao et al. (2019), Xiang |
|                                 | et al. (2019), S. Zhang et al.    |
|                                 | (2019), Sharma and Gupta (2018),|
|                                 | Babaei zadeh and Ahmad (2017),   |
|                                 | Yang and Dong (2016), Dhaliwal    |
|                                 | and Dhillon (2016), Xiang et al. |
|                                 | (2018), Kong et al. (2016), Zhou  |
|                                 | et al. (2016), Yang et al. (2016),|
|                                 | Wang and Zhang (2016), W. Mao     |
|                                 | et al. (2016), Zhao et al. (2016),|
|                                 | Worasucheep (2015), M. Li et al.  |
|                                 | (2015a), Bhandari et al. (2015),|
|                                 | Zhou et al. (2015a), F.-J. Kuang  |
|                                 | et al. (2015), T.-L. Li et al.    |
|                                 | (2015b), Zhou et al. (2015c),    |
|                                 | Zhao et al. (2015), Wang et al.   |
|                                 | (2015), Luo and Wang (2014), Sang |
|                                 | et al. (2015), F. Kuang et al.    |
|                                 | (2014), G. Li et al. (2014),     |
|                                 | Sharma and Pant (2013), Sharma et|
|                                 | al. (2013), Lai and Yu (2012), W.|
|                                 | Gao et al. (2012), El-Abd (2012),|
|                                 | Yang and Huang (2012), Gao et al.|
|                                 | (2012), X. Bi and Wang (2011a),  |
|                                 | Gao et al. (2011), Gao and Liu    |
|                                 | (2011), El-Abd (2011), Sharma    |
|                                 | and Pant (2011), X. J. Bi and     |
|                                 | Wang (2011b), Bao and Zeng (2011),|
|                                 | Kalaikumar and Baburaj (2020),    |
|                                 | Sharma and Abraham (2020), Shao    |
|                                 | et al. (2020), Liu et al. (2019),|
|                                 | Huang et al. (2019)               |
| **Memory and Archive**          | Amarjeet (2018a), Amarjeet (2018b),|
|                                 | Chhabra J.K. Amarjeet (2018c), X.\ |
|                                 | Li et al. (2019), Moradi et al.   |
|                                 | (2018), Zabihi and Nasiri (2018),|
|                                 | Fan et al. (2018), Du et al. (2017),|
|                                 | M. Mao et al. (2017), Mao and Duan|
|                                 | (2016), Yin and Chuang (2016), Li|
|                                 | and Yang (2016), Nakano et al.    |
|                                 | (2015), Parvin et al. (2018)     |
| **Others**                      | G. Wang et al. (2018a), Chowdhury |
|                                 | et al. (2013), Ren et al. (2019),|
|                                 | J. Wang et al. (2018b),          |
| Phases                  | Learning algorithms | Publications                                                                 |
|------------------------|---------------------|-------------------------------------------------------------------------------|
| Initialization         | Clustering          | Vigneswari and Mohamed (2016), Kumar and Sahoo (2017), Ibrahim et al. (2020), |
| OBL                    | Xiang et al. (2017), Y. Wang et al. (2017), Shameer and Subhajini (2019), S. Zhang et al. (2019), Babaeizadeh and Ahmad (2017), Kong et al. (2016), Yang et al. (2016), Wang and Zhang (2016), W. Mao et al. (2016), M. Li et al. (2015a), Bhandari et al. (2015), F.-J. Kuang et al. (2015), T.-L. Li et al. (2015b), Luo and Wang (2014), F. Kuang et al. (2014), G. Li et al. (2014), Lai and Qu (2012), W. Gao et al. (2012), El-Abd (2012), Gao et al. (2012), Gao et al. (2011), Gao and Liu (2011), El-Abd (2011), Bao and Zeng (2011), Kalaikumar and Baburaj (2020), Sharma and Abraham (2020), Liu et al. (2019), Huang et al. (2019)
| Others                 | Ren et al. (2019), J. Wang et al. (2018b), Employed Bees Clustering | Sun et al. (2015a), Sun et al. (2015b), Reinforcement Zhao and Zhang (2020), Fairee et al. (2019), Fairee et al. (2018), Ma and Zhang (2016), OBL Sharma and Gupta (2018), Yang and Dong (2016), Zhao et al. (2016), Wang (2015), Shao et al. (2020), Dhillon and Dhillon (2016), Worasueep (2015), Zhou et al. (2015a), Song et al. (2015), Sharma and Pant (2013), Sharma et al. (2013), Sharma and Pant (2011), Yang and Huang (2012)
| Memory and Archive     | Mao and Duan (2016), Others X. J. Wang et al. (2018b), OED Gao et al. (2013), Ren and Wu (2013), Onlooker Bees Clustering Karaboga and Gorkemli (2014), Gorkemli and Karaboga (2019), Gorkemli and Al-Dulaimi (2019), Aslan et al. (2019), Reinforcement Ma and Zhang (2016), OBL Sharma and Gupta (2018), Xiang et al. (2018), Dhillon and Dhillon (2016), Worasueep (2015), Zhou et al. (2015a), Zhao et al. (2015), Song et al. (2015), Sharma and Pant (2013), Sharma et al. (2013), Sharma and Pant (2011), Yang and Huang (2012), Sharma and Pant (2011), OED Gao et al. (2013), Ren and Wu (2013), Scout Bees Clustering Krishnamoorthi and Natarajan (2013), OBL Guo et al. (2019), Xiang et al. (2019), Zhou et al. (2016), Zhou et al. (2015c), Luo and Wang (2014), Lai and Qu (2012), Gao et al. (2012), X. Bi and Wang (2011a), X.-J. Bi and Wang (2011b), Shao et al. (2020), Zhao et al. (2015), El-Abd (2012), El-Abd (2011), OED X.-Y. Zhou et al. (2015b), C. Yang and Guo (2018), Memory and Archive Nakano et al. (2015), Population diversity control Clustering Moradi et al. (2018), Wen et al. (2020), Gao et al. (2015), Costa and De Oliveira (2013; 2014), Biswas et al. (2012), Sun and Chen (2016), Zhang et al. (2016), Zhang et al. (2017), Tran et al. (2020), Memory and Archive Amarjeet (2018a), Amarjeet (2018b), Chhabra J.K. Amarjeet (2018c), X. Li et al. (2019), Moradi et al. (2018), Fan et al. (2018), Du et al. (2017), M. Mao et al. (2017), Yin and Chuang (2016), Li and Yang (2016), Nakano et al. (2015), Parvin et al. (2018), Others Chowdhury et al. (2013), (Continued)
that employs ML algorithms for problem scaling and modelling objective function for increasing the performance of ABC in an application. These issues stand for the researchers dealing with the hybridization of ML and ABC for improving the performance of the ABC algorithm.

References

Abbass, H. A. (2001, May). Marriage in honey-bee optimization (MBO): A haplometrosis polygynous swarming approach. In The Congress on Evolutionary Computation (CEC2001) (pp. 207–214). Seoul, Korea.

Amarjeet, C. J. K. (2018a). TA-ABC: Two-archive artificial bee colony algorithm. Dervis Karaboga, Bahriye Akay & Nurhan Karaboga, Cogent Engineering (2021), 7: 1855741.

Amarjeet, C. J. K. (2018b). Many-objective artificial bee colony for multi-objective software module clustering problem. Soft Computing, 22(19), 6341–6361. https://doi.org/10.1007/s00500-017-2687-3

Amarjeet, C. J. K. (2018c). FP-ABC: Fuzzy-Pareto dominance driven artificial bee colony algorithm for many-objective software module clustering. Computer Languages, Systems and Structures, 51, 1–21. https://doi.org/10.1016/j.cl.2017.08.001

Aparak, M. K., Karaboga, D., & Akay, B. (2014). The artificial bee colony algorithm in layer optimization for the maximum fundamental frequency of symmetrical laminated composite plates. Engineering Optimization, 46(3), 420–437. https://doi.org/10.1080/0305215X.2013.776551

Aslan, S., Badem, H., & Karaboga, D. (2019). Improved quick artificial bee colony (iqABC) algorithm for global optimization. Soft Computing, 23(24), 13161–13182. https://doi.org/10.1007/s00500-019-03858-y

Babaeizadeh, S., & Ahmad, R. (2017). Enhanced constrained artificial bee colony algorithm for optimization problems. International Arab Journal of Information Technology, 14(2), 246–253. http://www.umc.edu.dz/images/Enhanced-Constrained-Artificial-Bee-Colony-Algorithm-for-Optimization-Problems.pdf

Bao, L., & Zeng, J.-C. (2011). A bi-group differential artificial bee colony algorithm. Kongzhi Lilun Yu Yingyong/ Control Theory and Applications, 28(2), 266–272.

Bezdek, J. C. (1981). Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, USA. ISBN 0-306-40671-3.

Bhandari, A. K., Kumar, A., & Singh, G. K. (2015). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur’s, Otsu and Tsallis functions. Expert Systems with Applications, 42(3), 1573–1601. https://doi.org/10.1016/j.eswa.2014.09.049

Bi, X., & Wang, Y. (2011a). An improved artificial bee colony algorithm. ICCRD2011-2011 3rd International Conference on Computer Research and Development, 2(5764108), 174–177. https://doi.org/10.1109/ICCRD.2011.5764108

Bi, X., & Wang, Y. (2011b). Artificial bee colony algorithm with fast convergence. Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics, 33(12), 2755–2761. https://doi.org/10.3969/j.issn.1001-506X.2011.12.34

Bishop, J. M. (1989). Stochastic searching networks. In Proceeding 1st IEE conference on artificial neural networks (pp. 329–331). London.

Biswas, S., Bose, D., & Kundu, S. (2012). A clustering particle based artificial bee colony algorithm for...
dynamic environment. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7677, 151–159. LNCS. https://doi.org/10.1007/978-3-642-35380-2_19

Chowdhury, A., Rakshit, P., Konar, A., & Janarthanan, R. (2013). An evolutionary approach for analysing the effect of interaction site structural features on protein- protein complex formation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8251, 656–661. LNCS. https://doi.org/10.1007/978-3-642-35380-2_19

Costa, T. S., & de Oliveira, A. C. M. (2015). Artificial bee and differential evolution improved by clustering search on continuous domain optimization. Soft Computing, 19(9), 2457–2468. https://doi.org/10.1007/s00500-015-1850-9

Costa, T. S., & de Oliveira, A. C. M. (2013). New clustering search approaches applied to continuous domain optimization. In 2013 IEEE Congress on Evolutionary Computation, CEC 2013, Cancun, Mexico. (pp. 3214–3220), art. no. 6557963. https://doi.org/10.1109/CEC.2013.6557963

Dedeturk, B. K., & Akay, B. (2020). Spam filtering using a genetic algorithm. In Proceedings of the 16th IEEE congress on evolutionary computation CEC.2007.4424885

Dhaliwal, K. K., & Dhillon, J. S. (2016). On the design and optimization of digital IIR filter using oppositional artificial bee colony algorithm. In 2016 IEEE Students’ Conference on Electrical, Electronics and Computer Science, SCECES 2016, Bhopal, India. (pp. 7509307). https://doi.org/10.1109/SCECES.2016.7509307

Dorigo, M., Maniezzo, V., & Colomi, A. (1996). Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29–41. https://doi.org/10.1109/3477.484436

Du, Z.-X., Liu, G.-Z., & Han, D.-Z. (2017). An improved artificial bee colony algorithm with memory. Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications, 40(5), 61–66. https://doi.org/10.13190/j.jbupt.2017-037

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 3(1), 32–57. 0022-0495. https://doi.org/10.1016/0022-0495(73)90029-0

El-Abd, M. (2011). Opposition-based artificial bee colony algorithm. In Genetic and Evolutionary Computation Conference, GECCO’11, Association for Computing Machinery, New York, NY, USA. (pp. 109–115). https://doi.org/10.1145/2001576.2001592

El-Abd, M. (2012). Generalized opposition-based artificial bee colony algorithm. In 2012 IEEE Congress on Evolutionary Computation, CEC 2012. Brisbane, QLD, Australia. (pp. 6252939). https://doi.org/10.1109/CEC.2012.6252939

Fairee, S., Khompatraporn, C., Prom-On, S., & Sirinoaovakul, B. (2018). Combinatorial artificial bee colony optimization with reinforcement learning updating for travelling salesman problem. In Proceedings of the 16th International Conference on Electrical Engineering/ Electronics,Computer, Telecommunications and Information Technology, ECTI-CON 2019, art. no. 8955176. Pattaya, Chonburi, Thailand, Thailand. (pp. 93–96). https://doi.org/10.1109/ECTI-CON4724&2019.8955176

Fairee, S., Prom-On, S., & Sirinoaovakul, B. (2018). Reinforcement learning for solution updating in artificial bee colony. PLoS ONE, 13(7), e0200738. art. no. e0200738. https://doi.org/10.1371/journal.pone.0200738

Fan, C., Fu, Q., Long, G., & Xing, Q. (2018). Hybrid artificial bee colony algorithm with variable neighborhood search and memory mechanism. Journal of Systems Engineering and Electronics, 29(2), 405–414. https://doi.org/10.21629/JSEE.2018.02.20

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial intelligence through simulated evolution. John Wiley & Sons.

Guo, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. Information Processing Letters, 111(17), 871–882. https://doi.org/10.1016/j.ipl.2011.06.002

Guo, W., Liu, S., & Huang, L. (2012). A global best artificial bee colony algorithm for global optimization. Journal of Computational and Applied Mathematics, 236(11), 2741–2753. https://doi.org/10.1016/j.cam.2012.01.013

Guo, W.-F., Huang, L.-L., Liu, S.-Y., & Dai, C. (2015). Artificial bee colony algorithm based on information learning. IEEE Transactions on Cybernetics, 45(12), 2827–2839. art. no. 7008482. https://doi.org/10.1109/TCYB.2014.2387067

Guo, W.-F., Liu, S.-Y., & Huang, L.-L. (2012). Inspired artificial bee colony algorithm for global optimization problems. Tien Tzu Hsueh Pao/Acta Electronica Sinica, 40(12), 2396–2403. https://doi.org/10.3969/j.issn.0372-2112.2012.12.007

Guo, W.-F., Liu, S.-Y., & Huang, L.-L. (2013). A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. IEEE Transactions on Cybernetics, 43(3), 1011–1024. https://doi.org/10.1109/TCYB.2012.2222373

Guo, W.-F., Liu, S.-Y., & Jiang, F. (2011). An improved artificial bee colony algorithm for directing orbits of chaotic systems. Applied Mathematics and Computation, 218(7), 3868–3879. https://doi.org/10.1016/j.amc.2011.09.034

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. Simulation, 76(2), 60–68. https://doi.org/10.1177/0037681101076002

Görkemli, B., & Al-Dulaimi, Z. (2019). On the performance of quick artificial bee colony algorithm for dynamic deployment of wireless sensor Networks. Turkish Journal of Electrical Engineering and Computer Sciences, 27(6), 4038–4054. https://doi.org/10.3906/elk-1802-389

Görkemli, B., & Karaboga, D. (2019). A quick semantic artificial bee colony programming (qsABCp) for symbolic regression. Information Sciences, 502, 346–362. https://doi.org/10.1016/j.ins.2019.06.052

Guo, Z., Shi, J., Xiong, X., Xia, X., & Liu, X. (2019). Chaotic artificial bee colony with elite opposition-based learning. International Journal of Computational Science and Engineering, 18(4), 383–390. https://doi.org/10.1504/IJCSSE.2019.099076

Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.

Hosseini, H. S. (2007). Problem solving by intelligent water drops. In IEEE congress on evolutionary computation (pp. 3226–3231). Singapore. https://doi.org/10.1109/CEC.2007.4424885

Huang, X., Zeng, X., Han, R., & Wang, X. (2019). An enhanced hybridized artificial bee colony algorithm for optimization problems. IAES International Journal of Artificial Intelligence, 8(1), 87–94. https://doi.org/10.11591/ijai.v8i1.p87-94

Ibrahim, Y. M., Darwish, S., & Sheta, W. (2020). Brain tumor segmentation in 3D-MRI based on artificial bee colony and level set. Advances in Intelligent Systems and Computing, 974, 491–500. https://doi.org/10.1007/978-3-319-91914-7_45

Ishibuchi, H., & Tanaka, H. (1991). A fuzzy satisficing decision making procedure under an interactive fuzzy multiattribute decision making procedure. Fuzzy Sets and Systems, 44(3), 249–265. https://doi.org/10.1016/0165-0114(91)90084-A

Jiang, F., & Qiao, Z. (2016). A novel chaotic artificial bee colony algorithm for combinatorial optimization problems based on chaotic modified PSO. International Journal of Modern Education and Computer Science, 8(5), 65–78. https://doi.org/10.7763/IJMECS.2016.8.5.6
Kara
to, D. (2005). An idea based on honey bee swarm for numerical optimization (Technical Report TR06).

Erciyes University, Engineering Faculty, Computer Engineering Department.

Kara
to, D., & Gorkemli, B. (2014). A quick artificial bee colony (aABC) algorithm and its performance on optimization problems. Applied Soft Computing Journal, 23, 227–238. https://doi.org/10.1016/j.asoc.2014.06.035

Kara
to, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2016). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42(1), 21–57. https://doi.org/10.1007/s10462-012-9328-0

Kara
to, D., & Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. Scientific Research and Essays, 5(14), 1899–1902. https://academici

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In 1995 IEEE International Conference on Neural Networks, 4, 1942–1948. https://doi.org/10.1109/INCN.1995.489968

Kirkpatrick, S. C., Gelatt, D., & Vecchi., M. P. (1983). Optimization by simulated annealing. Science, 220 (4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kong, X., Liu, S., Wang, Z., & Chang, X. (2016). An effective chaotic artificial bee colony approach to global optimization and its application. Journal of Computational and Theoretical Nanoscience, 13(5), 2878–2892. https://doi.org/10.1166/jctn.2016.4933

Koza, J. R. (1990). Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-13136, Stanford University Computer Science Department.

Krishnamoorthi, M., & Natarajan, A. M. (2013). Artificial bee colony algorithm integrated with fuzzy C-mean operator for data clustering. Journal of Computer Science, 9(4), 404–412. https://doi.org/10.3844/jcssp.2013.404.412

Kuang, F., Jin, Z., Xu, W., & Zhang, S. (2016). A novel chaotic artificial bee colony algorithm based on Tent map. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014. Beijing, China. (pp. 235–241), art. no. 6900278. https://doi.org/10.1109/CEC.2014.6900278

Kuang, F.-J., Jin, Z., Xu, W.-H., & Zhang, S.-Y. (2015). Hybridization algorithm of Tent chaos artificial bee colony and particle swarm optimization, Kongzhi Yu Juece/Control and Decision, 30(5), 839–847. https://doi.org/10.13195/j.kzyjc.2014.0750

Kumar, V., & Sahoo, G. (2017). A two-step artificial bee colony algorithm for clustering. Neural Computing & Applications, 28(3), 537–551. https://doi.org/10.1007/s00521-015-2095-5

Lai, L., & Qu, S. (2012). Path planning for unmanned air vehicles using an improved artificial bee colony algorithm. In Chinese Control Conference, CCC. Hefei, China. (pp. 2486–2491). art. no. 6390343.

Li, G., Niu, P., Ma, Y., Wang, H., & Zhang, W. (2014). Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency. Knowledge-Based Systems, 67, 278–289. https://doi.org/10.1016/j.knosys.2014.04.042

Li, M., Zhao, H., Wang, X., & Huang, H. (2015a). Artificial bee colony algorithm with comprehensive search mechanism for numerical optimization. Journal of Systems Engineering and Electronics, 26(3), 603–617. art. no. 7170019. https://doi.org/10.1109/JSEE.2015.00068

Li, T.-L., Liu, F.-A., & Wang, X.-H. (2015b). Modified artificial bee colony algorithm based on divide-and-conquer strategy. Kongzhi Yu Juece/Control and Decision, 30(2), 316–320. https://doi.org/10.13195/j.kzyjc.2013.1442

Li, X., & Yang, G. (2016). Artificial bee colony algorithm with memory. Applied Soft Computing Journal, 41, 362–372. https://doi.org/10.1016/j.asoc.2015.12.046

Li, X., Yang, H., Yang, M., & Yang, G. (2019). Flexible time-of-use tariff with dynamic demand using artificial bee colony with transferred memory scheme. Swarm and Evolutionary Computation, 46, 235–251. https://doi.org/10.1016/j.swevo.2019.02.006

Liu, P., Liu, X., Luo, Y., Du, Y., Fan, Y., & Feng, H.-M. (2019). An enhanced exploitation artificial bee colony algorithm based on automatic functional approximation and Intelligent Automation and Soft Computing, 25(2), 385–394. https://doi.org/10.31209/2019.0000100

Luo, J., & Wang, Q. (2014). A method for axis straightness error evaluation based on improved artificial bee colony algorithm. International Journal of Advanced Manufacturing Technology, 71(S–8), 1501–1509. https://doi.org/10.1007/s00170-013-5567-8

Ma, L., Wang, X., Huang, M., Lin, Z., Tian, L., & Chen, H. (2019). Two-level master-slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(5), 861–880. art. no. 7993093. https://doi.org/10.1109/TSMC.2017.2723483

Ma, P., & Zhang, H.-L. (2016). Improved artificial bee colony algorithm based on reinforcement learning. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9772, 721–732. https://doi.org/10.1007/978-3-319-42294-7_64

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(14), 281–297. https://projecteuclid.org/euclid.bsmsp/1200512992

Mao, M., & Duan, Q. (2016). Modified artificial bee colony algorithm with self-adaptive extended memory. Cybernetics and Systems, 47(7), 585–601. https://doi.org/10.1080/01969722.2016.1211881

Mao, M., Duan, Q., & Zhang, L. (2017). Artificial bee colony algorithm based on adaptive search equation and extended memory. Cybernetics and Systems, 48(5), 459–482. https://doi.org/10.1080/01969722.2017.1319240

Mao, W., Lan, H.-Y., Li, H.-R. (2016). New modified artificial bee colony algorithm with exponential function adaptive steps. Computational Intelligence and Neuroscience, 2016, 1–13. art. no. 9820294. https://doi.org/10.1155/2016/9820294

Moradi, M., Nejadian, S., Parvin, H., & Rezaie, V. (2018). CMABC: Clustering and Memory-Based Chaotic Artificial Bee Colony Dynamic Optimization Algorithm. International Journal of Information Technology & Decision Making, 17(6), 1007–1046. https://doi.org/10.1142/S0219622018500153

Nakano, H., Kojima, M., & Miyachi, A. (2015). An artificial bee colony algorithm with a memory scheme for...
