Journal of King Saud University – Computer and Information Sciences 32 (2020) 794–808

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Artificial bee colony algorithm: A component-wise analysis using diversity measurement

Kashif Hussain [a], Mohd Najib Mohd Salleh [a,*], Shi Cheng [b], Yuhui Shi [c], Rashid Naseem [d]

[a] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Johor, Malaysia
[b] School of Computer Science, Shaanxi Normal University, Xi'an, China
[c] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[d] Department of Computer Science, City University of Science and Information Technology, Peshawar, Pakistan

ARTICLE INFO

Article history:
Received 12 March 2018
Revised 27 August 2018
Accepted 21 September 2018
Available online 25 September 2018

Keywords:
Artificial bee colony
Metaheuristic
Exploration and exploitation
Component-wise analysis
Diversity measurement

ABSTRACT

A swarm-based metaheuristic algorithm, like artificial bee colony (ABC), embraces four key elements of collective intelligence: positive feedback, negative feedback, multiple interactions, and fluctuation. Fluctuation refers to population diversity which can be measured using dimension-wise diversity. This paper performed component-wise analysis of ABC algorithm using diversity measurement. The analysis revealed scout bees component as counterproductive and onlooker bees component with poor global search ability. Subsequently, an ABC algorithm without scout bees component and modified onlooker bees component is proposed in this paper. The effectiveness and efficiency of the proposed ScoutlessABC is validated on test suite of a dozen of benchmark functions. To further evaluate the performance, ScoutlessABC is employed on the parameter training problem of fuzzy neural network for solving eight classification problems. The experimental results show that ScoutlessABC maintains strong convergence ability than the original ABC algorithm. Overall, this study has two major contributions: (a) an effective component-wise analysis approach using diversity measurement and (b) a simplified and modified ABC variant with enhanced search efficiency.

1. Introduction

In metaheuristic research, swarm-based algorithms have earned significant place as compared to evolutionary and other population-based metaheuristic algorithms (Ab Wahab et al., 2015). There exist at-least over fifty swarm-based metaheuristic algorithms which have been successfully implemented in a wide variety of applications related to civil engineering, mechanical design, energy, transportation, electrical and electronics, business and economics, arts, etc. (Hussain et al., 2018). The collective intelligence of swarm individuals extracted from nature in the form of foraging behaviors of birds, ants, and bees, etc., has led to crafting efficient optimization techniques. Among such methods are classical swarm-based metaheuristic algorithms; such as, particle swarm optimization (PSO) (Eberhart and Kennedy, 1995), ant colony optimization (ACO) (Dorigo et al., 2006), and artificial bee colony (ABC) (Karaboga, 2005). Besides, some of the recently introduced swarm-based algorithms are bacterial foraging optimization (BFO) (Passino, 2012), chicken swarm optimization (CSO) (Meng et al., 2014), animal migration optimization (AMO) (Li et al., 2014), and crow search algorithm (CSA) (Askarzadeh, 2016).

When compared with classical and some of the recent methods, ABC has produced promising results. Hussain et al. (2017) found ABC as an efficient optimizer for fuzzy neural network while solving classification problems, as compared to PSO and mine blast algorithm (MBA) (Sadollah et al., 2013). ABC also generated significant results in clustering analysis problem when compared with AMO (Ma et al., 2015). Uymaz et al. (2015) and Anuar et al. (2016) verified effectiveness of ABC over bat algorithm (BA) (Yang, 2010), cuckoo search (CS) (Yang and Deb, 2009), and firefly algorithm (FA) (Yang, 2010) when implemented on several global optimization problems. In the domain of petroleum and oil-field operations, Nozohour-leilabady and Fazelabdolabadi (Nozohour-leilabady and Fazelabdolabadi, 2016) applied ABC and PSO on discovering optimal well locations. Results confirmed the proficiency of ABC over PSO. Garg (2014) applied ABC on various

* Corresponding author.
E-mail address: najib@uthm.edu.my (M.N. Mohd Salleh).

https://doi.org/10.1016/j.jksuci.2018.09.017
1319-1578/

structural engineering design problems, where researchers minimized the cost of design more effectively than other metaheuristic algorithms. Garg et al. (2013) proposed an efficient two-phased approach to solving reliable redundancy allocation problems. In this approach, ABC was applied on finding optimal solution of the reliability-redundancy allocation, which is further improved in the second phase. The ABC algorithm was employed on optimizing parameters of fuzzy logic-based industrial system. In comparison with conventional methods and other evolutionary algorithms, ABC achieved better results.

As ABC is an efficient and easy to implement algorithm, hence it is relatively more commonly used than other algorithms modeled over honey swarm; such as, mutable smart bee algorithm (MSBA) (Mozaffari et al., 2012), virtual bee algorithm (VBA) (Yang, 2005), honey bee colony algorithm (HBCA) (Chong et al., 2006), bee colony optimization (BCO) (Teodorovic and Dellrco, 2005), and bee swarm optimization (BSO) (Drias et al., 2005). Similar to other swarm-based metaheuristic algorithms, ABC also embraces key elements of swarm intelligence; i.e., positive feedback, negative feedback, multiple interactions, and fluctuation. ABC employs three different groups of bees to pursue the said features. Employed bees and onlooker bees endorse positive feedback and multiple interactions, whereas scout bees focus on the element of negative feedback and fluctuation. Fluctuation helps avoid trapping in sub-optimal locations by injecting diversity in population. Scout bees perform this task by bringing solutions from far-fetched *totally* random locations, in search process. Employed and onlooker bees also add randomness up-to some extent. Here, question arises that how much fluctuation or randomness is sufficient for efficient performance.

In literature, there has been introduced various modifications and hybrids of the ABC algorithm, in order to induce enough population diversity so that the trade-off balance between exploration and exploitation can be achieved. However, mostly the research related to ABC is inclined towards high-level experimental analysis (Bansal, 2018), as there are several questions yet to be addressed with practical evidence, such as:

- Why ABC suffers from poor exploitation ability?
- Why ABC occasionally stops proceeding to global optimum?
- Why ABC performance downgrades on complex optimization problems?

This study is an attempt to answer the core questions related to ABC performance analysis using component-wise diversity measurement. The core motivation of this study is an interesting research by Sörensen (2015) which entails the need of in-depth analysis, instead of merely high-level analysis, by deconstruction approach with component-view of metaheuristic algorithms. With this motive, this study is able to answer "*how and why it happened*" question related to ABC performance, as mostly literature is focused on shallow analysis related to "*what happened*" (Yang, 2012; Karaboga et al., 2014; Sörensen et al., 2018). As a result of in-depth practical analysis, this research is able to simplify ABC by eliminating scout bees component, as well as, effectively modifying onlooker bees component. Generally, this paper has twofold contributions: (a) practically illustrating useful component-wise analysis approach using population diversity measurement and (b) proposing simplified and efficient ABC variant.

The outline of the study is as follows. Section 2 briefs about ABC and the variants introduced by different researchers. Section 3 explains the methodology designed to achieve the objectives of this research. In this section, the components of ABC are theoretically analyzed using population diversity measurement. Based on analysis, ABC is improved in this section. For practical evidence of the proposed modification, experiments on test functions and classification problems are performed – detail of experiments is given in Section 4. Section 5 reports and discusses results with in-depth analysis. This empirical work is finally concluded in Section 6.

2. Artificial bee colony (ABC) algorithm and variants

ABC algorithm was introduced by Karaboga (Karaboga, 2005) which is based on how honeybees work together for collecting nectar from flowers. In ABC, the objective is to find the patch of flowers with maximum nectar amount (optimal solution). For achieving this, ABC divides the bee swarm into three groups: employed bees, onlooker bees, and scout bees. Each bee represents a D-dimensional solution, the bee which finds the best food source among all is most likely to be followed by other bees for converging to the best location.

Similar to PSO, ABC also utilizes the concept of memory for storing personal best location. A bee visits new location and later on compares with the best location it visited previously. If the new location is better, the old one is forgotten and the new location is memorized; otherwise, the memory remains unchanged. Initially, ABC starts with sending bees to random places. When returning back to beehive, the employed bees share information with onlooker bees which choose the employed bee to follow based on probability of selection (1):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (1)$$

where fit_i is objective function value (nectar amount) of a solution i and SN is the total number of food sources. Roulette- wheel selection method is performed on employed bees' probability values. The more is the amount of nectar shared by an employed bee, the more are the chances of selection by onlooker bees. The onlooker bee moves to new location v_i, guided by chosen employed bee, with the help of (2):

$$v_i = x_i + Rand_i(x_i - x_j) \quad (2)$$

where x_i is the current location in memory, x_j is the selected employed bee based on probability, and the randomness added to find nectar around location x_j. After multiple iterations, when any bee that is unable to find better food source for some time, it is replaced with scout bee v_{new}. The scout bee invoked for the failed bee roams around any random or unexplored region to explore the environment using (3):

$$v_{new} = lb_i + Rand_i(ub_i - lb_j) \quad (3)$$

where lb_i, ub_i, and $Rand_i$ is the lower bound, upper bound, and randomness between [0,1]. The next cycle starts again with employed bees which visit the neighborhoods of locations present in memory. The summarized three steps procedure of ABC is outlined in Algorithm 1 and depicted via flow chart given in Fig. 2a.

Original ABC was proposed for solving unconstrained optimization problems, however research related to ABC has produced many modifications and hybrids of the algorithm, to implement

Algorithm 1: ABC Procedure.

Initialize bee swarm

repeat

1. Place employed bees on flower patches to determine nectar amount
2. Move onlooker bees to the promising locations shared by employed bees
3. Replace poor bees with scout bees to find unexplored locations

until *Repeat until maximum iterations or stopping criteria*;

on wider variety of applications. Some of the recent ABC variants are highlighted as following.

Yurtkuran and Emel (2016) proposed ABC variant with enhanced explorative and exploitative capability. Instead of greedy selection between old and new solutions in original ABC, this variant employed solution acceptance rule and probabilistic multisearch method to allow ABC early explore the search space by using probability of worse solutions to be selected. With probabilistic multisearch strategy and new selection rule providing three alternate solutions to be selected based on predetermined probabilities, the balance between exploration and exploitation was achieved. Experimental results of well-known benchmark test functions, when compared with PSO, DE and their variants, proved improvement in ABC performance.

To improve the search pattern of employed and onlooker bees, Xu et al. (2013) proposed the use of a pool of good solutions to select from in employed bees and onlooker bees processes. With this change, the new solutions are generated around some of the best solutions in order to improve convergence speed of the algorithm. The new ABC (NABC) was compared with ABC and its variants on benchmark test functions, the results verified the enhanced search efficiency.

In another related work, Karaboga and Akay (2011) modified the ABC algorithm to solve problems with constraints and compared results with state-of-the-art metaheuristics including genetic algorithm, differential evolution, and PSO, etc. The variant made two changes in ABC: (a) modified parameter update equation of employed bee, and (b) used Deb's rules with tournament selection instead of greedy selection. The statistical analysis including ANOVA and ANOM suggested best suitable parameters for the variant. The same authors in another work (Akay and Karaboga, 2012) presented improved ABC with modifications in perturbation process while determining the number of parameters to be updated in new location. This variant overcame the problem of poor convergence of ABC on non-separable and composite constrained optimization problems. The results of experiments, when compared with the standard and variants of ABC as well as other popular population-based algorithms, suggested that the proposed modification improved ABC performance significantly.

In order to deal with another type of optimization problems called binary optimization, Ozturk et al. (2015) proposed genetic operators based ABC (GB-ABC) which modified initialization and employed bee equations by using crossover and swap operators. The robustness of the variant algorithm is verified by testing it on image clustering and 0–1 knapsack problem, as well as, on CEC2005 benchmark test functions. The comparisons with similar binary algorithms like binary PSO and binary GA proved the efficiency of the proposed algorithm.

Another important modification in ABC was recently proposed by Sharma et al. (2016), where lévy flight was employed as balanced randomization strategy to avoid extra exploration and poor convergence ability. In the variant, the lévy flight local search strategy was integrated as last (4th) step after employed bees step, onlooker bees step, and scout bee step.

Inspired from PSO, ABC algorithm was modified by Zhu and Kwong (2010) by incorporating the globally best bee (best solution found so far) information into position update Eq. (2). According to the paper, the modification improved exploitative capability of the algorithm and produced better results on benchmark test functions with low to high dimensions, as compared with the original ABC.

Qin et al. (2015) attempted to balance explorative and exploitative capabilities of ABC by integrating time varying strategy in order to vary the ratio between employed bees and onlooker bees with the passage of iterations. Depending on the nature of variation in ratio, linear or nonlinear, this paper proposed ABC with linear time varying strategy (LTVS), so called ABC-LTVS, as well as with nonlinear time varying strategy (NTVS), calling it ABC-NTVS. Testing performance on 21 benchmark test functions, the proposed ABC variants outperformed the standard ABC algorithm and the variant Gbest-guided ABC (GABC).

The modifications mentioned earlier proposed significant improvement in ABC algorithm, however none performed component-based analysis to examine the role of each component on the performance of the standard ABC algorithm. The research mentioned earlier claimed to have balanced exploration and exploitation of ABC algorithm, but no in-depth analysis is provided except for convergence graph and averaged end results of certain number of runs. Even though, such results and graphs provide enough prove of the end results; nevertheless, *how* and *why* the proposed modifications improved ABC performance and *why* the original algorithm performed poor is still practically and partially unknown. Furthermore, the modifications and enhanced strategies often supplement extra parameters and complexity to the algorithm. It is therefore significantly mandatory for any analytical research to perform in-depth and component-wise analysis in order to solve "black box" issue attached with metaheuristic performance. Hence, this research performed component-wise analysis of ABC algorithm to determine the effect of each component on search performance of the algorithm, the methodology is explained in subsequent section.

3. Methodology

The basic motivation of this current work is Sörensen's paper (Sörensen, 2015) which presented the idea of breaking metaheuristics into components so that a connection between components and the performance can be established. Moreover, Cheng et al. (2014) inspired this work to investigate swarm behavior by measuring diversity in population, it is discussed further in this

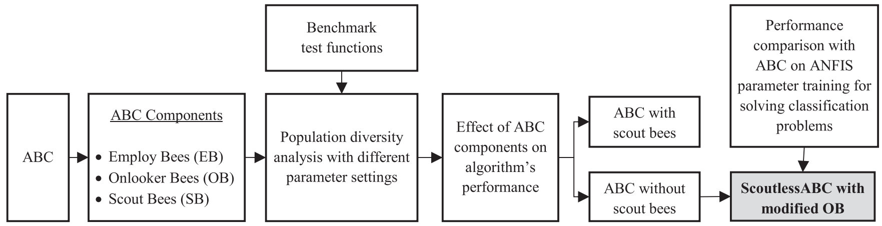


Fig. 1. Research methodology.

section. This helps empirically determine that how much fluctuation is suitable for efficient search.

The methodology adopted in this work is depicted in Fig. 1. As it is expressed via Fig. 1, ABC algorithm is broken into its three components: employed bees, onlooker bees, and scout bees components. The swarm agents created by these components individually are taken into account to measure dimension-wise diversity. This implied how effectively the search agents visit the environment – influencing exploration and exploitation capabilities of ABC algorithm. Based on hypothetical observation that scout bees generate solutions from randomly selected far-fetched locations, it makes this component highly fluctuating. Hence, an ABC algorithm without scout bees component was tested. Furthering the possibilities of improvement, modification on onlooker bees component was made and validated against the standard ABC.

The ABC algorithm is deconstructed in the following subsection which theoretically analyzes components.

3.1. ABC components

Based on the research gap discussed earlier, it is highly imperative to analyze ABC algorithm from component-view. Since, ABC algorithm divides its workforce into three groups: employed bees (EB), onlooker bees (OB), and scout bees (SB), these groups can be viewed as components of ABC algorithm. Theoretically speaking, EB component is responsible for focusing search in the neighborhoods already present in memory. OB component takes care of exploiting potential neighborhoods, whereas SB component is to generate random solutions from far-fetched unseen neighborhoods to foster exploration in ABC.

From the view of collective intelligence, EB and OB components implement the two key elements: positive feedback and multiple interactions. The rest of the two elements, negative feedback and fluctuation are implemented by SB component in ABC algorithm. Fluctuation or randomness is also incorporated in the swarm by EB and OB components as *Rand* in (2). The scale of randomness is often controlled by a coefficient which can be set to a higher or lower value for increased or limited diversity. In case of SB component, the level of diversity is significantly high since solutions are generated completely randomly as compared to EB and OB components.

The appearance of SB component in the search process can be controlled by the *Limit* parameter. Nevertheless, scout bees are used to replace any existing solutions which have failed to improve for a number of times; however, these solutions can be potential or the best found so far. Generally, based on implementations of SB in literature with $Limit = SN \times D$, where SN is swarm size and D is problem dimensions, SB component appears towards the later part of search process when ABC is about to converge to an already identified potential neighborhood. At this stage, the introduction of completely random solutions may disrupt convergence and other swarm agents may choose neighborhoods of scout bees which are far from the neighborhoods of employed and onlooker bees. This may introduce population diversity, but the level of diversity in this case will be unnecessarily high that it may diverge search from already identified potential locations in the environment. It can be inferred that population diversity is a useful element of fluctuation in search process, but unnecessarily high diversity may inversely affect the search efficiency of a metaheuristic algorithm. For better and adequate diversity, acceleration coefficient of EB and OB components can be adjusted accordingly. Based on the theoretical analysis presented above, this work first tried elimination of scout bees component and verified performance on optimization problems, also validated in comparison with standard ABC algorithm. Later, this work modified OB component.

In the original ABC algorithm, the position update Eq. (2) of OB component is based on distance from the selected potential employed bee location x_j with the perspective of current position x_i, that is $(x_i - x_j)$. When observed particle swarm optimization (PSO) and firefly algorithm (FA)(Yang, 2009), it is reverse. The distance is calculated from the perspective of the selected potential location x_j, hence it should be $(x_j - x_i)$. Therefore, we proposed modification in the position update equation of OB component as (4):

$$v_i = x_j + Rand_i(x_j - x_i) \tag{4}$$

where x_i is current position and x_j is the selected promising location found by employed bees.

To summarize, the proposed modification in standard ABC algorithm includes the elimination of scout bees component and modification in OB component. The flowchart of standard ABC and the proposed ScoutlessABC algorithm is presented via Fig. 2. As depicted in Fig. 2b, the ScoutlessABC algorithm was simplified by elimination of the SB component and modification in the OB component as well.

3.2. Population diversity measurement

In any swarm-based metaheuristic algorithm, the diversity of solutions offered by swarm individuals can be measured through its dimensions. As each swarm individual represents D-dimensional vector of parameters to be optimized, the large difference between dimensions implies that the swarm individuals are placed at distant locations. Inversely, in case of close distance between dimensions, the swarm individuals are converging or focusing on specific neighborhood in the search space. For the measurement of dimension-wise diversity of each swarm individual and whole as a swarm, we considered the equation proposed in Cheng et al. (2014) with modification as in (5):

$$\begin{aligned} Div_j &= \frac{1}{n}\sum_{i=1}^{n}\left|median(x_j) - x_{ij}\right|, \\ Div &= \frac{1}{D}\sum_{i=1}^{n} Div_j, \end{aligned} \tag{5}$$

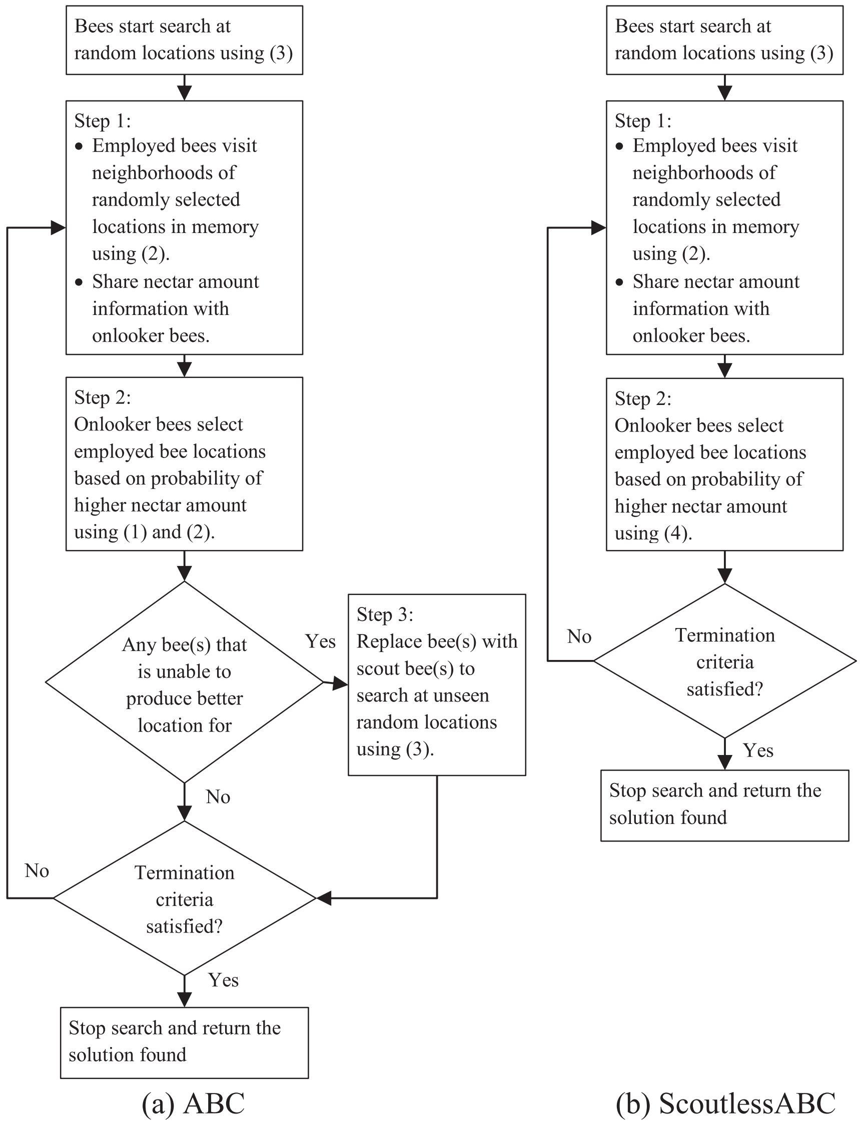


Fig. 2. Flowcharts of standard ABC and the proposed ScoutlessABC algorithms.

where $median(x_j)$ is median of dimension j in whole swarm, whereas x_{ij} is the dimension j of bee i and n is the total number of bees. When measuring diversity of each bees group, n represents number of bees in that particular group. After taking dimension-wise distance of each bee i from the median of dimension j, we take average Div_j of the bees.

With diversity measurement in hand, it is now possible to determine exploration and exploitation abilities of the ABC algorithm. To calculate exploration and exploitation in each iteration of search process, (6) can be used:

$$\begin{aligned} Exploration &= \frac{Div}{Div_{max}} \times 100, \\ Exploitation &= \left(1 - \frac{Div}{Div_{max}}\right) \times 100, \end{aligned} \tag{6}$$

where Div is the diversity measurement of whole swarm in an iteration, Div_{max} is the maximum diversity of swarm in all iterations.

It is noteworthy to mention that exploration and exploitation are two corner-stones of metaheuristic performance. Exploration is discovering new neighborhoods in search space, whereas exploitation is furthering search in already identified potential neighborhoods. Any metaheuristic algorithm with high exploration ability may make long jumps in search space and avoid potential solutions, whereas with high exploitation ability it may suffer from lack of diversity in solutions and trap in local optimum locations. Measuring these two features may significantly help maintain the trade-off balance between exploration and exploitation or make effective modifications in the algorithm where necessary.

The theoretical analysis and improvement in the algorithm discussed above are further validated via experiments performed in the upcoming section and analyses are presented in results and discussion section.

4. Experiments

In order to achieve better comprehension of the theoretical analysis presented in this work and to obtain sufficient practical evidence, we performed experiments on commonly used benchmark test functions, as well as, on classification problems. We aimed at providing not only end results but also to present sound illustrations to investigate swarm behavior, in terms of population diversity and measurement of exploration and exploitation, during the process of search for an optimal solution. Hence, the influence of each component of the ABC algorithm, i.e., employed bees,

Table 1
Numerical optimization problems with $D = 30$.

Test Function	Equation	Range	Optimum
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
Step	$f_2 = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]^D$	0
SumSquares	$f_3 = \sum_{i=1}^{D} i x_i^2$	$[-10, 10]^D$	0
Quartic	$f_4 = \sum_{i=1}^{D} i x_i^4 + \text{Random}(0, 1)$	$[-1.28, 1.28]^D$	0
Schwefel 1.2	$f_5 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$	0
Schwefel 2.22	$f_6 = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$	0
Dixon and Price	$f_7 = (x_1 - 1)^2 \sum_{i=1}^{D} i (2x_i^2 - x_{i-1})^2$	$[-10, 10]^D$	0
Rosenbrock	$f_8 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	0
Rastrigin	$f_9 = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	0
Ackley	$f_{10} = -20 exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	$[-32.768, 32.768]^D$	0
Griewank	$f_{11} = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$	0
Penalized	$f_{12} = \frac{\pi}{D} \times \left\{ 10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \times [1 + \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ where $y_i = 1 + 0.25(x_i + 1)$ and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leqslant x_i \leqslant a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	0

onlooker bees, and scout bees, was recorded to present as results. The results are reported in this study in terms of diversity measurement, exploration and exploitation percentage, and best fitness value found in maximum iterations.

In order to test the search efficiency of ABC and the ScoutlessABC, twelve commonly used benchmark numerical optimization test functions with different modalities were employed as test-bed. The functions along with characteristics and equations are given in Table 1. In our test-bed, unimodal functions are from f_1 to f_7 and from f_8 to f_{12} are multimodal functions in nature.

For further comprehensive analysis and stress testing on high dimensional problems, we evaluated ABC and ScoutlessABC also on real-world optimization problems, apart from benchmark numerical optimization problems. Unlike numerical optimization problems, training of Adaptive Neuro-Fuzzy Inference System (ANFIS) (Jang, 1993) parameters for solving classification problems is highly complex and non-convex problem. Hence, performing component-wise analysis on such problems helps comprehend swarm behavior of ABC on real-life problems. Since, ABC and ScoutlessABC algorithms were employed on training ANFIS parameters (membership function parameters and consequent parameters) for solving five classification problems, hence the feature length of the dataset determines the problem dimensions. Table 2 lists datasets, taken from UCI (University of California Irvin) machine learning repository[1] and KEEL (Knowledge Extraction based on Evolutionary Learning) repository[2], which include different range of features. Accordingly, our classification problems dimensions range from 39 to around 60000 (Table 3).

Briefly, ANFIS (Jang, 1993) is a neural network type machine learning algorithm, which involves fuzzy logic and inference mechanism to generate fuzzy rules. It is a five-layer architecture in which first layer is membership function layer, second layer performs product operation on fuzzy membership degrees to compute rule strength, third layer normalizes a rule's strength against all the rules in ANFIS architecture, fourth layer is simple polynomial equation that computes output of each rule and it contains $n + 1$ trainable parameters, and lastly fifth layer simply aggregates the outputs of all the rules to generate single output. The parameters in first and fourth layers are trainable. Originally, ANFIS trains these parameters using gradient-based method, instead in our experiments, we used ABC and ScoutlessABC algorithms to train ANFIS parameters. The greater detail of ANFIS can be found in Jang (1993). In this experimental study, we used grid partitioning method to generate m^n rules where m is the number of membership functions per input and n are inputs. For each input, we used three membership functions of Gaussian shape which uses two parameters. To calculate total trainable parameters, or in other words problem dimensions, following formula can be used (7):

Table 2
Classification problems.

Dataset ID	Dataset Name	Features	Instances
D_1	Iris	4	150
D_2	Banana	2	5300
D_3	Haberman's Survival	3	306
D_4	Post-Operative Patient	8	90
D_5	Hayes-Roth	5	160

$$\begin{aligned} N_r &= m^n, \\ N_p^{MF} &= n \times m \times 2, \\ N_p^{Con} &= N_r \times (n + 1), \\ N_p^T &= N_p^{MF} + N_p^{Con} \end{aligned} \tag{7}$$

where N_r, N_p^{MF}, N_p^{Con}, and N_p^T are total rules in ANFIS architecture, total membership function parameters, total consequent parameters of fourth layer of ANFIS architecture, and total trainable parameters or problem dimensions. Table 2 lists dimension size associated with each dataset. Accordingly, minimum dimension-size is 39 for Banana dataset and maximum dimensions reach up to around 60000 for Post-Operative Patient dataset, which form significantly large and complex optimization problem.

Since experiments were performed on two different kinds of problems: numerical optimization problems and ANFIS parameters training for classification problems, the ABC and ScoutlessABC algorithms employed separate parameter settings given in Table 4. For numerical optimization problems, the algorithms were run 30

[1] https://archive.ics.uci.edu/ml/datasets.html.
[2] http://sci2s.ugr.es/keel/datasets.php.

Table 3
ANFIS training complexity (problem dimensions).

Dataset	Inputs	MFs per Input	Rules	Premise Parameters	Consequent Parameters	Total Parameters (Dimensions)
D1	4	3	81	24	405	429
D2	2	3	9	12	27	39
D3	3	3	27	18	108	126
D4	8	3	6561	48	59049	59097
D5	5	3	243	30	1458	1488

*MF = Membership function.
*Gaussian MF is used with two parameters (center and width).

Table 4
Parameter settings for ABC and ScoutlessABC for experiments.

Parameters	Values	
ABC/ScoutlessABC Settings		
	Numerical Problems	Classification Problems
Swarm size	50	50
α	0.6	0.6
Limit	$SN \times D$	50
Lower and upper bounds	Refer Table 1	[-10,10]
Maximum iterations	2000	50
ANFIS Settings		
Membership function type	Gaussian	
Membership functions per input	3	
Rule generation method	Grid partitioning	

times and 10 times for classification problems. The results presented in the following section are averaged over specified runs.

5. Results and discussion

Since the objective of this study is to investigate contribution of each ABC component in overall algorithm's performance, we measured diversity and fitness values of swarm agents in the related components separately during iterations. Overall performance of the algorithm is also reported so that comparison of the proposed ScoutlessABC can be made with the standard ABC. The experimental results of numerical optimization problems, as well as, training of fuzzy neural network for solving classification problems have been presented in this section. Furthermore, a Friedman test at 0.05 significance level is also performed to statistically validate significance of the proposed modification.

Component-wise diversity measurement and fitness values obtained on numerical problems are presented in Table 5 whereas results of classification problems are presented in Table 6. Furthermore, for illustrative evidence of swarm behavior, Figs. 3–14 present component-wise diversity of ABC and ScoutlessABC on numerical problems, whereas Figs. 15–26 provide exploration and exploitation measurement during search process on numerical problems; on the other hand, Fig. 27 shows convergence abilities of ABC and ScoutlessABC on numerical problems. The results of classification problems are shown via Figs. 28–32 for component-wise diversity measurement. Figs. 33–37 demonstrate exploration and exploitation abilities of ABC and ScoutlessABC on classification problems, whereas convergence of ABC and ScoutlessABC on classification problem are illustrated via Figs. 38.

According to component-wise diversity provided in Table 5, the diversity of SB component is significantly higher than EB and OB components in all twelve test functions, which shows that SB component is extra-ordinarily divergent as it generates solutions completely randomly far from EB and OB components. In all the test functions, SB component in original ABC achieved average diversity over 6 which is extravagantly higher than maximum average diversity 0.09 achieved by EB component and 0.22 by OB component. When the random solutions generated by SB component are incorporated in EB component in the next iteration, an unnecessarily high diversity is injected into the swarm. This disrupts convergence of swarm to already identified potential neighborhoods. With this in view, the SB component was eliminated in the proposed ScoutlessABC algorithm. Another modification in ScoutlessABC was a small but useful change in the position update Eq. (4) of OB component. The modified OB component did not make much difference in diversity, however it focused on more promising regions in search environment. This change also affected EB component in ScoutlessABC as it was guided by agents located in global optimum locations, as compared to the same component in ABC.

ScoutlessABC achieved significantly better global optimum solutions as compared to ABC. In fact, for two test functions, one unimodal function Step (f_2) and one multimodal function

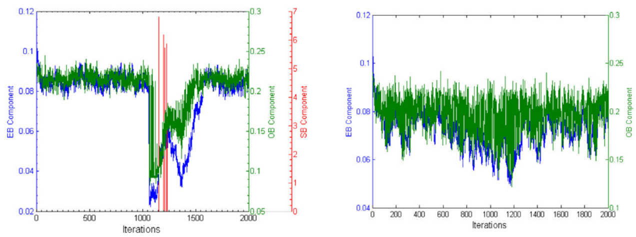
Table 5
Component-wise diversity and performance on numerical optimization problems.

Fun.	ABC					ScoutlessABC			
	EB	OB	SB	Swarm	Fitness	EB	OB	Swarm	Fitness
f_1	0.0758	0.2021	6.1382	0.0803	9.30E−16	0.0713	0.1977	0.0738	8.88E−45
f_2	0.0792	0.2048	6.6040	0.0846	2.92E−15	0.0881	0.2360	0.0977	0.00E+00
f_3	0.0765	0.2040	6.5716	0.0811	1.61E−14	0.0901	0.2374	0.1008	4.74E−39
f_4	0.0868	0.2222	6.2738	0.0979	7.72E−03	0.0896	0.2316	0.1029	4.63E−03
f_5	0.0866	0.2168	6.2591	0.0939	6.60E+01	0.0622	0.1909	0.0654	9.61E−03
f_6	0.0649	0.1825	6.4268	0.0654	5.87E−07	0.0838	0.2264	0.0912	7.31E−21
f_7	0.0799	0.2054	6.2591	0.0853	6.67E−01	0.1812	0.1812	0.0772	6.67E−01
f_8	0.0741	0.2040	6.3098	0.0784	2.76E+01	0.0456	0.1074	0.0458	2.47E+01
f_9	0.0891	0.2166	5.7014	0.0983	1.91E+02	0.0600	0.1735	0.0650	2.89E+01
f_{10}	0.0796	0.2067	6.0782	0.0851	3.68E−08	0.0912	0.2394	0.1029	6.22E−15
f_{11}	0.0626	0.1670	6.5644	0.0640	4.08E−06	0.0517	0.1750	0.0582	0.00E+00
f_{12}	0.0811	0.2090	6.1280	0.0887	3.38E−02	0.0356	0.1407	0.0362	1.57E−32

*EB = Employed bees component, OB = Onlooker bees component, SB = Scout bees component.

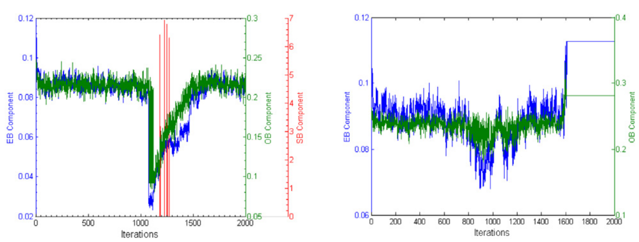
Table 6
Component-wise diversity and performance on classification problems.

Dataset	ABC					ScoutlessABC			
	EB	OB	SB	Swarm	MSE	EB	OB	Swarm	MSE
D_1	1.1534	2.9286	90.1870	1.1883	0.0034	1.2523	3.3240	1.3567	0.0015
D_2	0.1083	0.2734	68.2350	0.1130	0.0034	0.3793	0.9907	0.4170	0.0014
D_3	0.3486	0.8846	26.1463	0.3629	0.0013	0.3625	0.9626	0.3887	0.0015
D_4	1.1765	2.9871	52.8222	1.2196	0.0277	1.2766	3.3730	1.4017	0.0221
D_5	1.1840	2.9863	52.5076	1.2374	0.0326	1.2650	3.3360	1.3854	0.0145



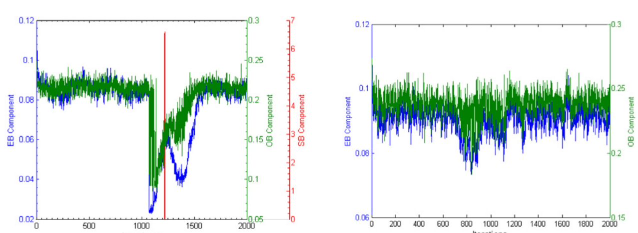
(a) ABC (b) ScoutlessABC

Fig. 3. Component-wise diversity measurement on Sphere (f_1).



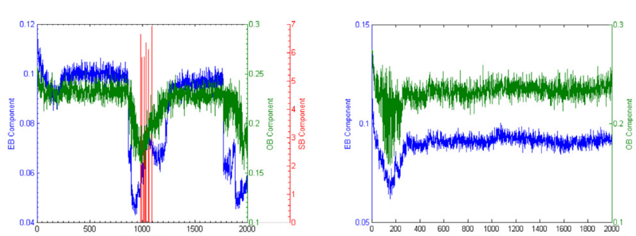
(a) ABC (b) ScoutlessABC

Fig. 4. Component-wise diversity measurement on Step (f_2).



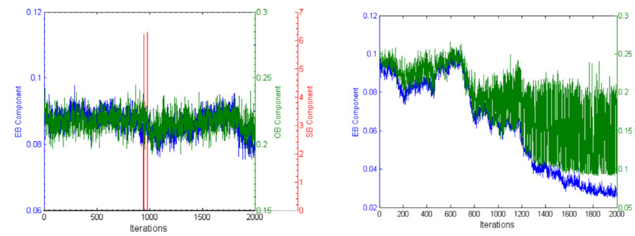
(a) ABC (b) ScoutlessABC

Fig. 5. Component-wise diversity measurement on SumSquare (f_3).



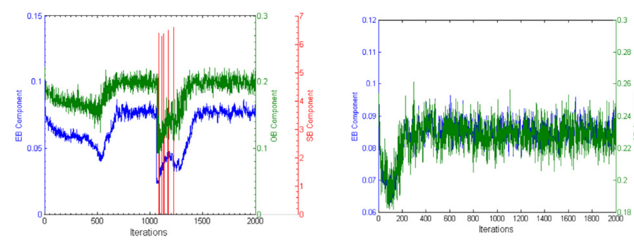
(a) ABC (b) ScoutlessABC

Fig. 6. Component-wise diversity measurement on Quartic (f_4).



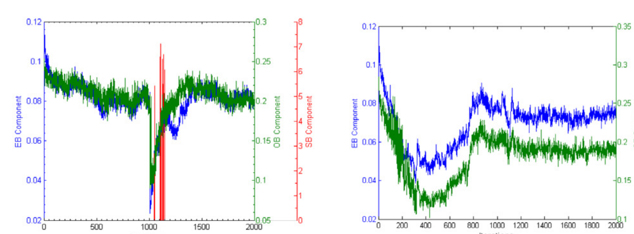
(a) ABC (b) ScoutlessABC

Fig. 7. Component-wise diversity measurement on Schwefel 1.2 (f_5).



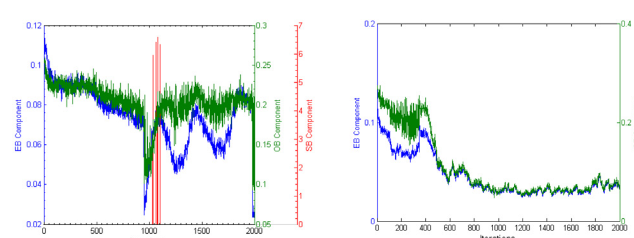
(a) ABC (b) ScoutlessABC

Fig. 8. Component-wise diversity measurement on Schwefel 2.22 (f_6).



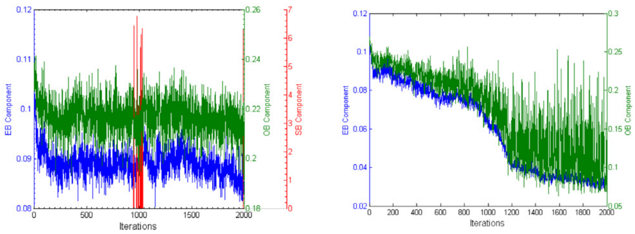
(a) ABC (b) ScoutlessABC

Fig. 9. Component-wise diversity measurement on Dixon and Price (f_7).



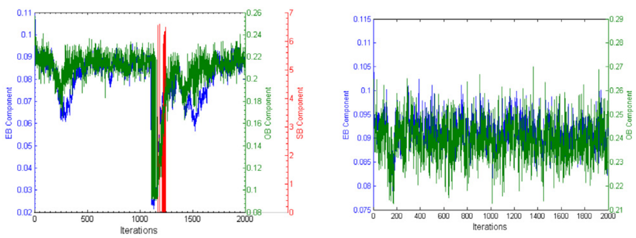
(a) ABC (b) ScoutlessABC

Fig. 10. Component-wise diversity measurement on Rosenbrock (f_8).



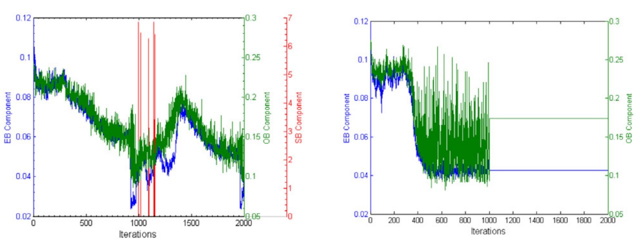
(a) ABC (b) ScoutlessABC

Fig. 11. Component-wise diversity measurement on Rastrigin (f_9).



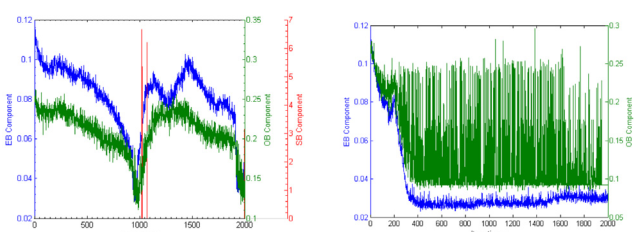
(a) ABC (b) ScoutlessABC

Fig. 12. Component-wise diversity measurement on Ackley (f_{10}).



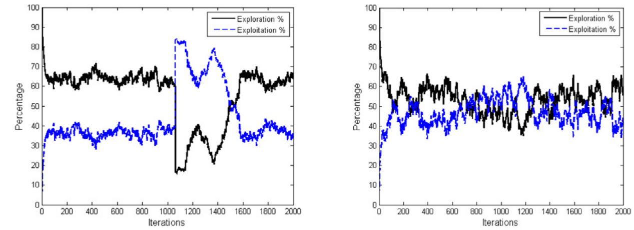
(a) ABC (b) ScoutlessABC

Fig. 13. Component-wise diversity measurement on Griewank (f_{11}).



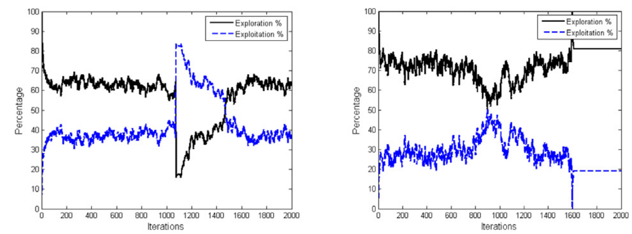
(a) ABC (b) ScoutlessABC

Fig. 14. Component-wise diversity measurement on Penalized (f_{12}).



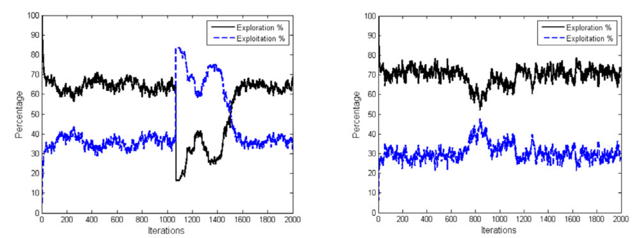
(a) ABC (b) ScoutlessABC

Fig. 15. Exploration and exploitation measurement on Sphere (f_1).



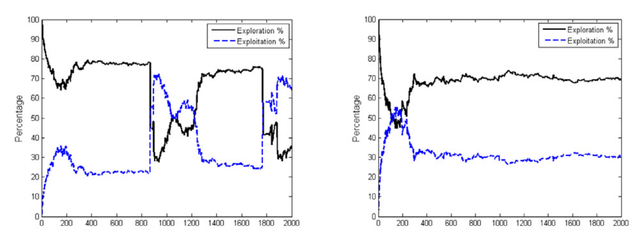
(a) ABC (b) ScoutlessABC

Fig. 16. Exploration and exploitation measurement on Step (f_2).



(a) ABC (b) ScoutlessABC

Fig. 17. Exploration and exploitation measurement on SumSquares (f_3).

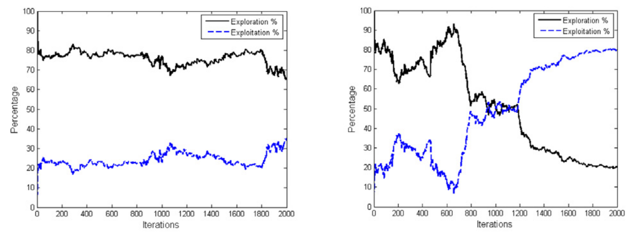


(a) ABC (b) ScoutlessABC

Fig. 18. Exploration and exploitation measurement on Quartic (f_4).

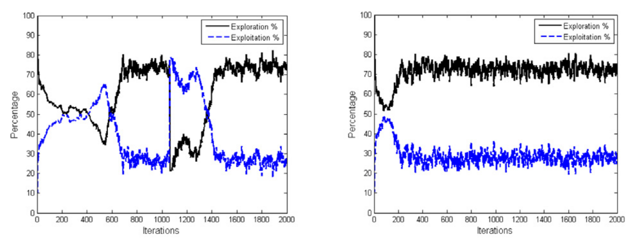
Griewank (f_{11}), ScoutlessABC achieved the desired global optimum value (0). Whereas, ABC hardly solved Step (f_2) with suboptimal solution 2.92E−15 and Griewank (f_{11}) with solution 4.08E−06. Other robust performances by ScoutlessABC were for Sphere (f_1), SumSquares (f_3), and Penalized (f_{12}) which is considered as hard optimization problem. ScoutlessABC generated optimum solutions 8.88E−45, 4.74E−39, and 1.57E−32 respectively for f_1,f_3, and f_{12} as compared to solutions 9.30E−16, 1.61E−14, and 3.38E−02 generated by ABC respectively for the same functions. However, there are problems where ScoutlessABC performed as equally better as ABC; i-e, Quartic (f_4), Dixon and Price (f_7), and Rosenbrock (f_8). Overall, the significance of ScoutlessABC can be validated by statistical tools like Friedman Test. The p-value = 0.0164 obtained by Friedman Test with significance level 0.05 suggests that there is significant difference between the results of ScoutlessABC and ABC. ScoutlessABC achieved considerably better results as compared to ABC.

Knowing the end results in terms of diversity or objective function values may not fully help understand the swarm behavior unless it is not revealed that how swarm behaved during the course of iterations. For that reason, this study illustratively explains swarm behavior in standard ABC and ScoutlessABC using



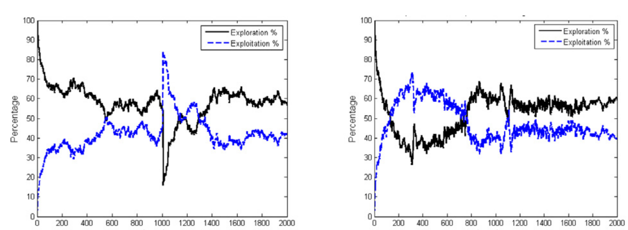
(a) ABC (b) ScoutlessABC

Fig. 19. Exploration and exploitation measurement on Schwefel 1.2 (f_5).



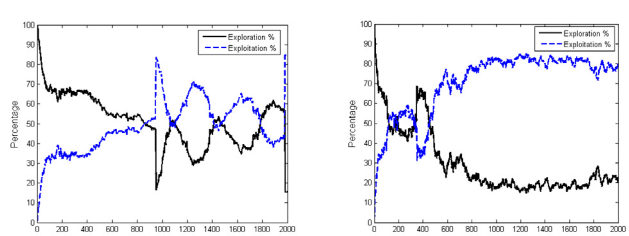
(a) ABC (b) ScoutlessABC

Fig. 20. Exploration and exploitation measurement on Schwefel 2.22 (f_6).



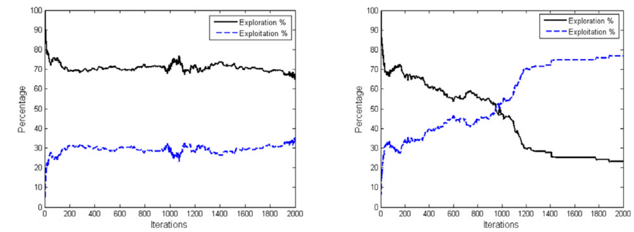
(a) ABC (b) ScoutlessABC

Fig. 21. Exploration and exploitation measurement on Dixon and Price (f_7).



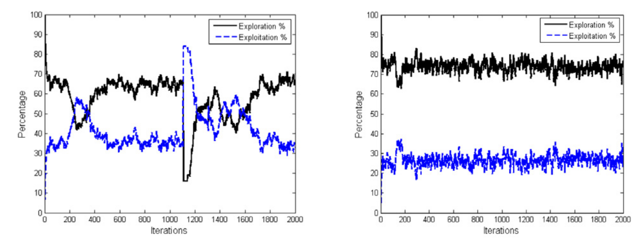
(a) ABC (b) ScoutlessABC

Fig. 22. Exploration and exploitation measurement on Rosenbrock (f_8).



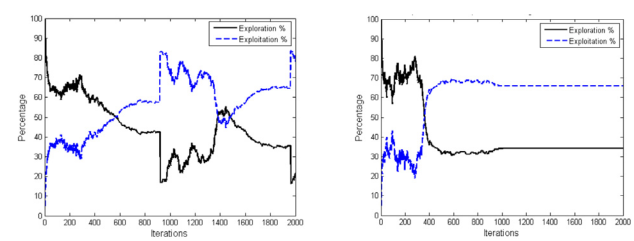
(a) ABC (b) ScoutlessABC

Fig. 23. Exploration and exploitation measurement on Rastrigin (f_9).



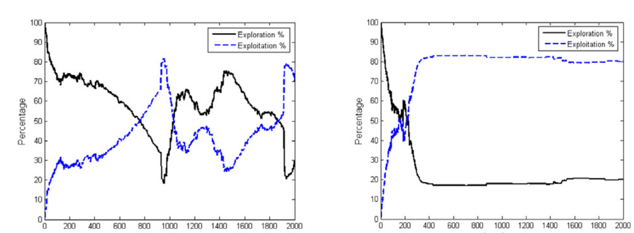
(a) ABC (b) ScoutlessABC

Fig. 24. Exploration and exploitation measurement on Ackley (f_{10}).



(a) ABC (b) ScoutlessABC

Fig. 25. Exploration and exploitation measurement on Griewank (f_{11}).

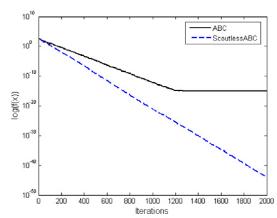


(a) ABC (b) ScoutlessABC

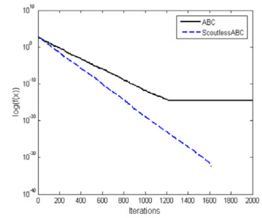
Fig. 26. Exploration and exploitation measurement on Penalized (f_{12}).

diversity measurement and exploration and exploitation measurement during iterations. From the graphical evidence provided by Figs. 3a–14a showing component-wise diversity and measurement during 2000 iterations, there are three major deliberations that can be made about the ABC algorithm: (a) in most of the test functions, diversity in OB component remained higher than EB component during search process; (b) SB component in ABC mostly appeared in middle of the search process with significantly high diversity (reaching up to 7) as compared to EB and OB components; (c) the appearance of SB component in middle of the search process created a jolt in the diversity of EB and OB components, which indicates disruption in convergence.

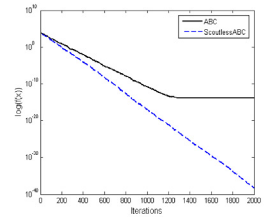
The diversity measurement of the proposed ScoutlessABC is illustrated by Figs. 3b–14b. It is obvious from graphs that the absence of SB component in ScoutlessABC helped avoid any disruption in convergence, as opposite to ABC. Moreover, the proposed modification in OB component made it more vibrant throughout iterations in most of the test functions. For example, OB component in ScoutlessABC was specially more dynamic than in ABC on Sphere (f_1), Quartic (f_4), Schwefel 1.2 (f_5), Rastrigin (f_9), and Penalized (f_{12}).



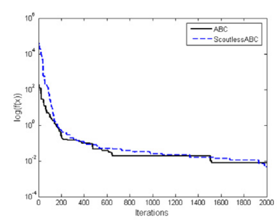
(a) Sphere (f_1)



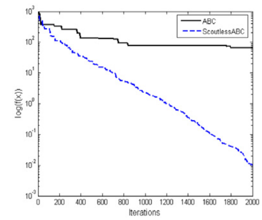
(b) Step (f_2)



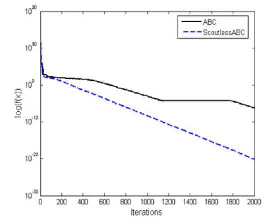
(c) SumSquares (f_3)



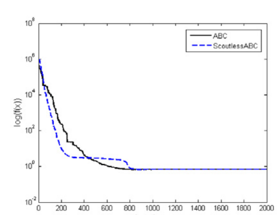
(d) Quartic (f_4)



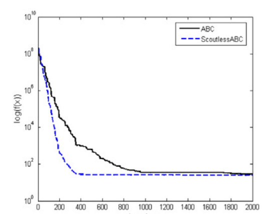
(e) Schwefel 1.2 (f_5)



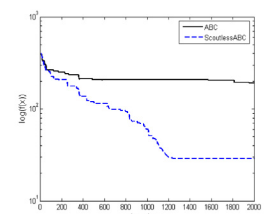
(f) Schwefel 2.22 (f_6)



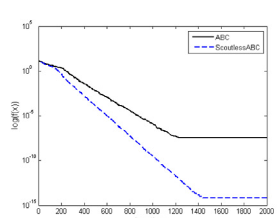
(g) Dixon and Price (f_7)



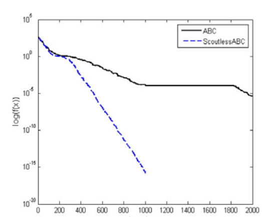
(h) Rosenbrock (f_8)



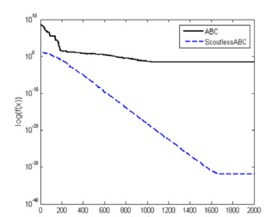
(i) Rastrigin (f_9)



(j) Ackley (f_{10})

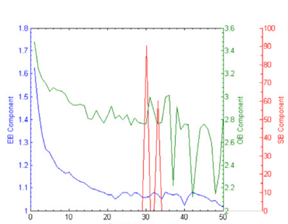


(k) Griewank (f_{11})

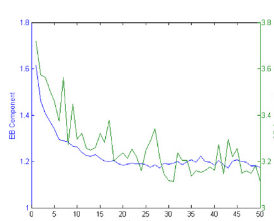


(l) Penalized (f_{12})

Fig. 27. Convergence of ABC and ScoutlessABC on numerical optimization problems.

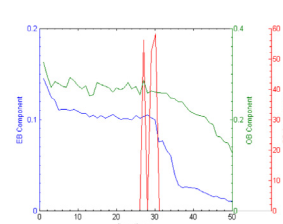


(a) ABC

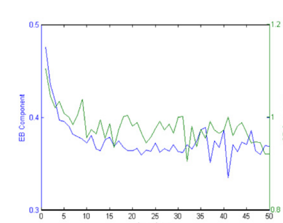


(b) ScoutlessABC

Fig. 28. Component-wise diversity measurement on Iris (D_1).

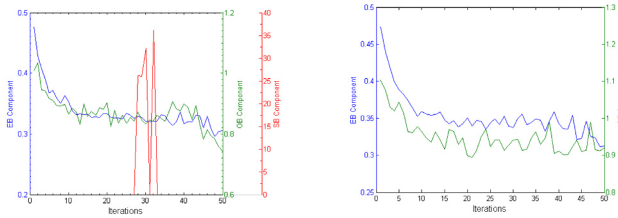


(a) ABC



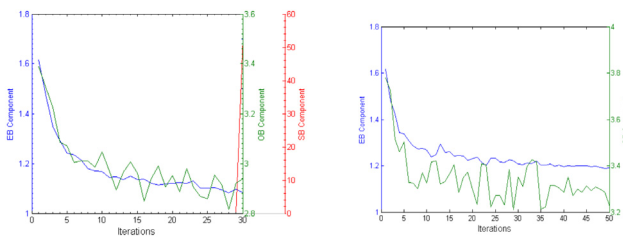
(b) ScoutlessABC

Fig. 29. Component-wise diversity measurement on Banana (D_2).



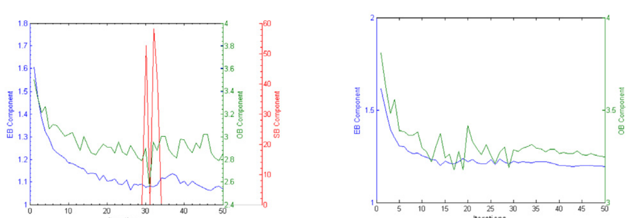
(a) ABC (b) ScoutlessABC

Fig. 30. Component-wise diversity measurement on Haberman's Survival (D_3).



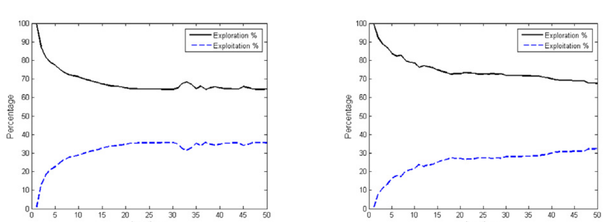
(a) ABC (b) ScoutlessABC

Fig. 31. Component-wise diversity measurement on Post-Operative Patient (D_4).



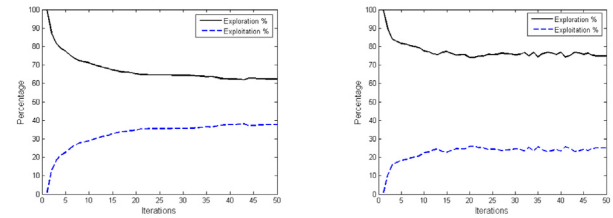
(a) ABC (b) ScoutlessABC

Fig. 32. Component-wise diversity measurement on Hayes-Roth (D_5).



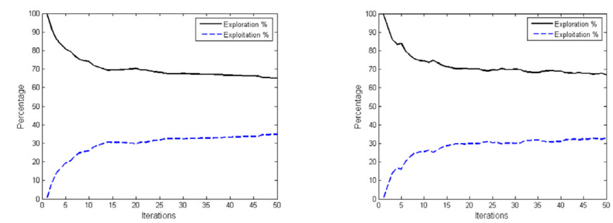
(a) ABC (b) ScoutlessABC

Fig. 33. Exploration and exploitation measurement on Iris (D_1).



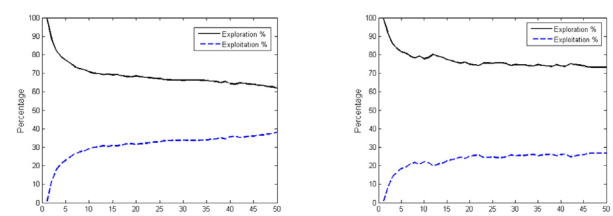
(a) ABC (b) ScoutlessABC

Fig. 34. Exploration and exploitation measurement on Banana (D_2).



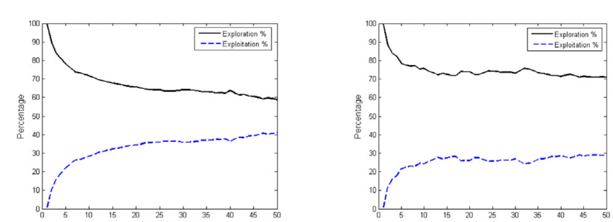
(a) ABC (b) ScoutlessABC

Fig. 35. Exploration and exploitation measurement on Haberman's Survival (D_3).



(a) ABC (b) ScoutlessABC

Fig. 36. Exploration and exploitation measurement on Post-Operative Patient (D_4).

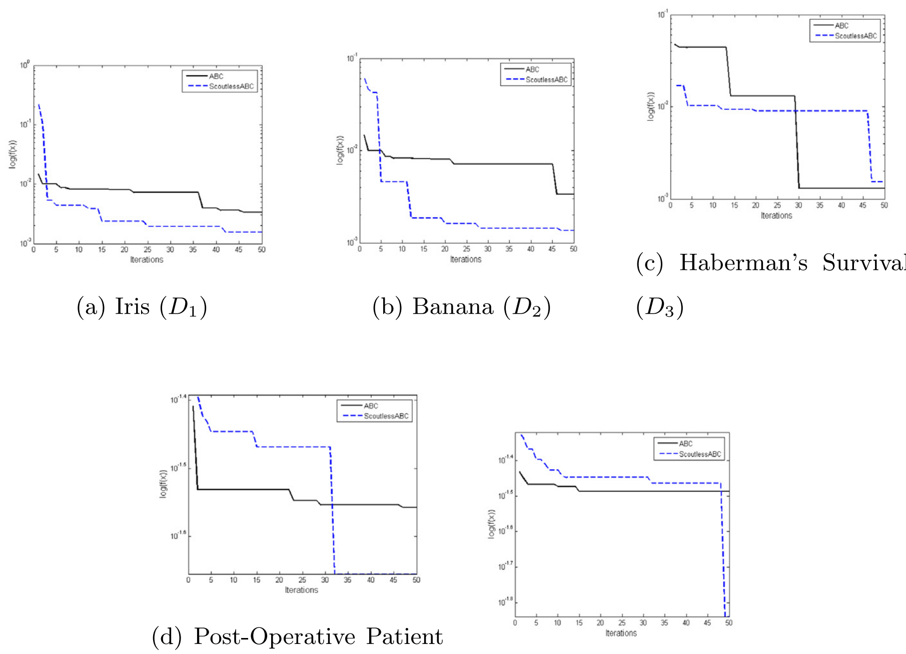


(a) ABC (b) ScoutlessABC

Fig. 37. Exploration and exploitation measurement on Hayes-Roth (D_5).

The deliberations made earlier about ABC and ScoutlessABC are further validated via exploration and exploitation graphs provided via Figs. 15–26. It is obvious that abnormal jumps in exploration and exploitation indicate appearance of SB component in middle of the search process of ABC. The fluctuations suggest that the SB component in ABC created disruption in the smooth convergence. On the other hand, the graphs clearly show relatively smooth exploration and exploitation ratios during iterations in ScoutlessABC. For example, Figs. 15b, 17b, 20b, and 24b for Sphere (f_1), SumSquares (f_3), Schwefel 2.22 (f_6), and Ackley (f_{10}), respectively, show that exploration and exploitation ratios remained consistent during most part of search process in ScoutlessABC.

The results of proposed elimination of SB component and effective modification in OB component are clearly evident in convergence ability of ScoutlessABC on numerical optimization problems. According to convergence graphs provided via Figs. 27 show that ScoutlessABC converged efficiently on global optimum locations, as compared to ABC. Other than test functions Quartic (f_4) and Dixon and Price (f_7), ScoutlessABC converged to global optimum locations much faster than ABC.



(a) Iris (D_1)

(b) Banana (D_2)

(c) Haberman's Survival (D_3)

(d) Post-Operative Patient (D_4)

(e) Hayes-Roth (D_5)

Fig. 38. Convergence of ABC and ScoutlessABC on classification problems.

The ABC and ScoutlessABC algorithms were also implemented on the parameter training problem of ANFIS for solving five classification problems. Results of experiments are averaged over 10 independent runs and presented in Table 6. According to component-wise mean diversity measurement of ABC, OB component was more vibrant than EB component while solving classification problems, except for Banana (D_2). The mean diversity measurement in EB component remained within the range of 0.1083 and 1.1840 as compared to OB component diversity which ranged from 0.2734 to 2.9871. Moreover, as in numerical problems, SB component of ABC behaved similarly in classification problems as well. The diversity measurement of SB component was significantly higher than EB and OB components, ranging from 26.15 to 90.19. This much diversity of SB component is enough to diverge swarm from convergence to already identified potential neighborhood. To cater this, the elimination of SB component was proposed in ScoutlessABC. Furthermore, to enhance convergence ability to global optimum locations, the modification in OB component worked positively in ScoutlessABC. As shown in Table 6, the modification in OB component of ScoutlessABC enhanced its diversity as compared to OB in ABC, while solving classification problems. The mean diversity measurement of OB component in ScoutlessABC remained within the range of 0.9626 and 3.3730, which is higher than OB in ABC. However, diversity in EB component of ScoutlessABC remained almost same as in ABC. As far as error measure is concerned, the proposed ScoutlessABC achieved smaller mean squared error (MSE) on four out of five classification problems as compared to ABC. ABC obtained MSE of 0.0013 as compared to 0.0015 achieved by ScoutlessABC on Haberman's Survival (D_3) problem. The results are statistically validated by Friedman test (Table 7). Statistically, there is no significant difference between the performances of ABC and ScoutlessABC on classification problems; however, smaller difference in MSE is considered as improvement in classification accuracy. Therefore, it can be inferred that ScoutlessABC achieved better accuracy on classification problems as compared to ABC, while training ANFIS parameters.

Table 7
Statistical results of Friedman test (ABC vs. ScoutlessABC).

	Numerical optimization problems	Classification problems
p-value	0.0164	0.2453

The results of diversity measurement of ABC and ScoutlessABC on classification problems are illustrated via Figs. 28–32. It is clearly visible that the diversity of OB component in ScoutlessABC remained higher than in ABC during most part of iterations. Furthermore, it can also be observed the SB component appeared in later part of iterations in ABC, with significantly high diversity. This may inversely affect convergence of ABC algorithm when the swarm is about to gather around already identified potential neighborhood. From the exploration and exploitation graphs provided in Figs. 33–37, it can be observed that in comparison with ABC, ScoutlessABC remained relatively more explorative than exploitative on all five classification problems, mainly due to modification in OB component of ScoutlessABC. Fig. 38 shows convergence ability of ABC and ScoutlessABC on classification problems. According to graphs, ScoutlessABC showed better convergence ability as compared to ABC on most of the classification problems.

From the detailed results of comprehensive experiments performed in this study, following observations can be made:

- Component-wise analysis using diversity measurement provided practical evidence of the role of employed bees, onlooker bees, and scout bees in the search efficiency of the ABC algorithm. Moreover, this approach also helped understand swarm behavior on optimization problems with wider range of complexities.
- The theoretical analysis suggested that employed bees and onlooker bees components are responsible for focusing on already identified regions in search space, whereas scout bees component is to perform global search in order to farsee the promising regions in search space. However, empirical analysis

suggested that onlooker bees component can also perform global search if modified suitably. Moreover, practical analysis also revealed that scout bees component is inversely effective to search efficiency of the ABC algorithm. Due to appearance in the later part of search, with significantly high diversity, scout bees component causes disruption in convergence of swarm to already identified promising neighborhood. Because it injects completely random solutions in the search process, it may misguide other two components of ABC by unnecessary divergence from already identified neighborhood.

- Component-wise analysis revealed that scout bees are highly divergent, which inject significantly high diversity in swarm that it disrupts swarm convergence to already identified potential neighborhood. The scout bees component can either be eliminated from ABC algorithm or modified to generate diversity in accordance with search progress, instead of blindly replacing a solution with completely random solutions.
- The measurement of exploration and exploitation during iterations also suggested that the ratio of exploration and exploitation is suddenly fluctuated due to introduction of scout bees during several times in search process of the ABC algorithm. There needs to be smooth transition from exploration to exploitation mode in order to let the swarm collectively decide about promising regions. The sudden introduction of scout bees during search process is like induction of stranger in a team which is already about to reach solution.
- For controlling diversity in swarm, acceleration coefficient attached with randomization is a useful parameter. It can be adaptively adjusted to inject adequate diversity in accordance with search status. Moreover, this parameter can also be incorporated with OB component to adjust diversity according to search progress.

6. Conclusions

This paper deconstructed ABC algorithm into its components to investigate the role of each component in performance efficiency, using diversity measurement. With diversity measurement in hand, we gauged exploration and exploitation in the algorithm. According to empirical evidence in the form of results of experiments on benchmark test functions, it can be concluded that the performance of ABC algorithm is inversely affected by scout bees component which produces totally random solutions regardless of the progress of search process. During experiments, it was noticed that employed bees and onlooker bees components already perform search effectively via well coordinated effort. The appearance of scout bees in the later part of iterations not only disrupts swarm momentum but also damages convergence ability of the algorithm, and the algorithm loses control of trail towards the possible globally optimum location. The elimination of scout bees component from standard ABC algorithm not only solved the issue of disruption in search process but also simplified algorithm structure. For further improvement, this study proposed modification in onlooker bees component which enhanced convergence ability of ABC algorithm. The component-wise analytical study based on diversity measurement can readily be applied on other population based algorithms. Other than the limited test functions employed in this study, a wider range of test functions and real-world applications may be used in future studies to analyze true performance efficiency of metaheuristic algorithms, using the proposed analytical approach.

Considering three important questions asked earlier in this research, following answers could be established from the comprehensive analysis:

- ABC suffers from poor exploitation because of scout bees component which appears in middle of the search process and causes disruption in convergence.
- ABC occasionally stops proceeding to global optimum locations because completely random solutions are injected into the search process by scout bees, which may unnecessarily diverge swarm from already identified promising neighborhood.
- ABC performance on complex optimization problems can be upgraded with the help of improvement in onlooker bee component by enhancing its global searchability.

References

Ab Wahab, M.N., Nefti-Meziani, S., Atyabi, A., 2015. A comprehensive review of swarm optimization algorithms. PLoS One 10 (5), e0122827.

Akay, B., Karaboga, D., 2012. A modified artificial bee colony algorithm for real-parameter optimization. Inf. Sci. 192, 120–142.

Anuar, S., Selamat, A., Sallehuddin, R., 2016. A modified scout bee for artificial bee colony algorithm and its performance on optimization problems. J. King Saud Univ.-Comput. Inf. Sci. 28 (4), 395–406.

Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput. Struct. 169, 1–12.

Bansal, J.C., Gopal, A., Nagar, A.K., 2018. Stability analysis of artificial bee colony optimization algorithm. Swarm Evol. Comput.

Cheng, S., Shi, Y., Qin, Q., Zhang, Q., Bai, R., 2014. Population diversity maintenance in brain storm optimization algorithm. J. Artif. Intell. Soft Comput. Res. 4 (2), 83–97.

Chong, C.S., Sivakumar, A.I., Low, M.Y.H., Gay, K.L., 2006. A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 38th conference on Winter simulation, Winter Simulation Conference. pp. 1954–1961.

Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. IEEE Comput. Intell. Mag. 1 (4), 28–39.

Drias, H., Sadeg, S., Yahi, S., 2005. Cooperative bees swarm for solving the maximum weighted satisfiability problem. Comput. Intell. Bioinspired Syst., 417–448

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, pp. 39–43.

Garg, H., 2014. Solving structural engineering design optimization problems using an artificial bee colony algorithm. J. Indus. Manage. Optimiz. 10 (3), 777–794.

Garg, H., Rani, M., Sharma, S., 2013. An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique. Comput. Oper. Res. 40 (12), 2961–2969.

Hussain, K., Salleh, M.N.M., Cheng, S., Shi, Y., 2017. Comparative analysis of swarm-based metaheuristic algorithms on benchmark functions. In: International Conference in Swarm Intelligence. Springer, pp. 3–11.

Hussain, K., Salleh, M.N.M., Cheng, S., Shi, Y., 2018. Metaheuristic research: a comprehensive survey. Artif. Intell. Rev., 1–43

Jang, J.-S., 1993. Anfis: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23 (3), 665–685.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga, D., Akay, B., 2011. A modified artificial bee colony (abc) algorithm for constrained optimization problems. Appl. Soft Comput. 11 (3), 3021–3031.

Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2014. A comprehensive survey: artificial bee colony (abc) algorithm and applications. Artif. Intell. Rev. 42 (1), 21–57.

Li, X., Zhang, J., Yin, M., 2014. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput. Appl. 24 (7–8), 1867–1877.

Ma, M., Luo, Q., Zhou, Y., Chen, X., Li, L., 2015. An improved animal migration optimization algorithm for clustering analysis. Discrete Dyn. Nat. Soc.

Meng, X., Liu, Y., Gao, X., Zhang, H., 2014. A new bio-inspired algorithm: chicken swarm optimization. In: International conference in swarm intelligence. Springer, pp. 86–94.

Mozaffari, A., Gorji-Bandpy, M., Gorji, T.B., 2012. Optimal design of constraint engineering systems: application of mutable smart bee algorithm. Int. J. Bio-Inspired Comput. 4 (3), 167–180.

Nozohour-leilabady, B., Fazelabdolabadi, B., 2016. On the application of artificial bee colony (abc) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (pso) methodology. Petroleum 2 (1), 79–89.

Ozturk, C., Hancer, E., Karaboga, D., 2015. A novel binary artificial bee colony algorithm based on genetic operators. Inf. Sci. 297, 154–170.

Passino, K.M., 2012. Bacterial foraging optimization. In: Innovations and Developments of Swarm Intelligence Applications. IGI Global, pp. 219–234.

Qin, Q., Cheng, S., Zhang, Q., Li, L., Shi, Y., 2015. Artificial bee colony algorithm with time-varying strategy. Discrete Dyn. Nat. Soc.

Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M., 2013. Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13 (5), 2592–2612.
Sharma, H., Bansal, J.C., Arya, K., Yang, X.-S., 2016. Lévy flight artificial bee colony algorithm. Int. J. Syst. Sci. 47 (11), 2652–2670.
Sörensen, K., 2015. Metaheuristics the metaphor exposed. Int. Trans. Oper. Res. 22 (1), 3–18.
Sörensen, K., Sevaux, M., Glover, F., 2018. A History of Metaheuristics. Handbook of Heuristics.
Teodorovic, D., DellOrco, M., 2005. Bee colony optimization – a cooperative learning approach to complex transportation problems. Advanced OR and AI methods in transportation. pp. 51–60.
Uymaz, S.A., Tezel, G., Yel, E., 2015. Artificial algae algorithm (aaa) for nonlinear global optimization. Appl. Soft Comput. 31, 153–171.
Xu, Y., Fan, P., Yuan, L., 2013. A simple and efficient artificial bee colony algorithm. Math. Problems Eng. 2013, 9.
Yang, X.-S., 2005. Engineering optimizations via nature-inspired virtual bee algorithms. Artif. Intell. Knowl. Eng. Appl.: A Bioinspired Approach, 317–323.
Yang, X.-S., 2009. Firefly algorithms for multimodal optimization. In: International Symposium on Stochastic Algorithms. Springer, pp. 169–178.
Yang, X.-S., 2010. A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, pp. 65–74.
Yang, X.-S., 2010. Firefly algorithm, levy flights and global optimization. In: Research and Development in Intelligent Systems XXVI. Springer, pp. 209–218.
Yang, X.-S., 2012. Nature-inspired mateheuristic algorithms: success and new challenges. arXiv preprint arXiv:1211.6658.
Yang, X.-S., Deb, S., 2009. Cuckoo search via lévy flights. In: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, pp. 210–214.
Yurtkuran, A., Emel, E., 2016. An enhanced artificial bee colony algorithm with solution acceptance rule and probabilistic multisearch. Comput. Intell. Neurosci. 2016, 41.
Zhu, G., Kwong, S., 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl. Math. Comput. 217 (7), 3166–3173.