Balanced artificial bee colony algorithm

Jagdish Chand Bansal* and Harish Sharma

ABV-Indian Institute of Information Technology
and Management, Gwalior,
Morena Link Road, Gwalior Madhya Pradesh, 474015 India
Fax: +91-751-2449813
E-mail: jcbansal@gmail.com
E-mail: harish.sharma0107@gmail.com
*Corresponding author

Atulya Nagar

Centre for Applicable Mathematics and
Systems Science (CAMSS),
Department of Computer and
Mathematical Sciences, Liverpool Hope University,
Liverpool L16 9JD, UK
E-mail: nagara@hope.ac.uk

K.V. Arya

ABV-Indian Institute of Information Technology
and Management, Gwalior,
Morena Link Road, Gwalior Madhya Pradesh, 474015 India
Fax: +91-751-2449829
E-mail: kvarya@gmail.com

Abstract: Artificial bee colony (ABC) optimisation algorithm is relatively a recent and simple population-based probabilistic approach for global optimisation over continuous and discrete spaces. It has reportedly outperformed a few evolutionary algorithms (EAs) and other search heuristics when tested over both benchmark and real world problems. ABC, like other probabilistic optimisation algorithms, has inherent drawback of premature convergence or stagnation that leads to the loss of exploration and exploitation capability of ABC. Therefore, in order to find a trade-off between exploration and exploitation capability of ABC algorithm two modifications are proposed in this paper. First, a new control parameter namely, cognitive learning factor (CLF) is introduced in the employed bees phase and onlooker bees phase. Cognitive learning is a powerful mechanism that adjusts the current position of candidate solution by a means of some specified knowledge. Second, the range of ABC control parameter ϕ is modified. The proposed strategy named as balanced artificial bee colony (*BABC*) algorithm, balances the exploration and exploitation capability of the ABC. To prove efficiency of the algorithm, it is tested over 24 benchmark problems of different complexities and compared with the basic *ABC*.

Keywords: numerical optimisation; artificial bee colony algorithm; cognitive learning factor; CLF; swarm intelligence.

Reference to this paper should be made as follows: Bansal, J.C., Sharma, H., Nagar, A. and Arya, K.V. (2013) 'Balanced artificial bee colony algorithm', *Int. J. Artificial Intelligence and Soft Computing*, Vol. 3, No. 3, pp.222–243.

Biographical notes: Jagdish Chand Bansal is an Assistant Professor at ABV-Indian Institute of Information Technology and Management, Gwalior. He obtained his PhD in Mathematics from IIT Roorkee. He is the Editor-In-Chief of *International Journal of Swarm Intelligence* (*IJSI*) published by Inderscience. His primary area of interest is nature inspired optimisation techniques.

Harish Sharma received his BTech and MTech degrees in Computer Engineering from Govt. Engineering College, Kota and Rajasthan Technical University, Rajasthan in 2003 and 2009, respectively. He is currently a Research Scholar at ABV-Indian Institute of Information Technology and Management, Gwalior, India.

Atulya Nagar is the Foundation Professor of Computer and Mathematical Sciences at Liverpool Hope University and is the Head of the Department of Computer Science. He obtained his PhD in Applied Non-linear Mathematics from the University of York (UK) in 1996. A mathematician by training, he brings multi-disciplinary expertise in computational science, bioinformatics, operations research and systems engineering to the Faculty of Business and Computer Sciences. He has published a substantial number of research papers in reputed outlets such as the IEE and IEEE publications.

K.V. Arya is working as an Associate Professor at ABV-Indian Institute of Information Technology and Management, Gwalior, India. He obtained his PhD in Computer Science and Engineering from Indian Institute of Technology (IIT Kanpur), Kanpur, India. He has more than 20 years of experience to teach the undergraduate and postgraduate classes. He has published more than 75 journal and conference papers in the area of information security, image processing, biometrics, wireless ad hoc networks and soft computing.

1 Introduction

Swarm intelligence is a meta-heuristic approach in the field of nature inspired techniques that is used to solve optimisation problems. It is based on the collective behaviour of social creatures. Social creatures utilises their ability of social learning to solve complex tasks. Researchers have analysed such behaviours and designed algorithms that can be used to solve non-linear, non-convex, or combinatorial optimisation problems in many science and engineering domains. Previous research (Dorigo and Stutzle, 2004; Kennedy and Eberhart, 1995; Price et al., 2005; Vesterstrom and Thomsen, 2004) have shown that algorithms based on swarm intelligence have great potential to find a solution of real world optimisation problem. The algorithms that have emerged in recent years include ant colony optimisation (ACO) (Dorigo and Stutzle, 2004), particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995), bacterial foraging optimisation (BFO) (Passino, 2002), artificial bee colony (ABC) optimisation (Karaboga, 2005), etc. Exploration and exploitation are the important mechanisms in a robust search process. While exploration process is related to the independent search for an optimal solution, exploitation uses existing knowledge to bias the search. In the recent years, there are few algorithms based on bee foraging behaviour developed to improve both exploration and exploitation capability for solving the numerical optimisation problems (Haijun and Qingxian, 2008; Banharnsakun et al., 2010b; Gao and Liu, 2011). The ABC algorithm introduced by Karaboga (2005) is one approach that has been used to find an optimal solution of numerical optimisation problems. This algorithm is inspired by the behaviour of honey bees when seeking a quality food source. ABC scheme is relatively a simple, fast and population-based stochastic search technique.

A lot of developments, comparative studies and applications of ABC have been carried out in recent years. Performance of the ABC algorithm has been compared with genetic algorithm (GA) (Goldberg, 1989), differential evolution (DE) (Storn and Price, 1995; Price, 1996; Price et al., 2005), particle swarm inspired evolutionary algorithm (PS-EA) (Karaboga and Basturk, 2007), PSO and evolutionary algorithm (EA) (Karaboga and Basturk, 2008). ABC, DE and PSO algorithms were studied for measuring the effect of search space scaling in Akay and Karaboga (2008). In Karaboga and Basturk (2007), Karaboga presented an extended version of ABC for constrained optimisation problems. He applied it to train neural networks (Karaboga et al., 2007), to medical pattern classification and clustering problems (Akay et al., 2008) and to solve TSP problems (Xing et al., 2007). Banharnsakun et al. (2010a) applied the ABC algorithm on distributed environments. Comparative performance analysis of ABC algorithm for automatic voltage regulator (AVR) system are carried out by Taplamacioglu and Gozden (2011). Singh (2009) applied the ABC algorithm on the leaf-constrained minimum spanning tree (LCMST) problem (ABC-LCMST) further he compared the approach with ACO and GA. In Singh (2009), comparison of ABC-LCMST is carried out, in terms of the best, average objective function value and computational time is carried out. It was observed that ABC-LCMST outperforms the other approaches. Mala et al. (2010) applied the ABC optimisation-based approach in automated software test optimisation framework. Dahiya et al. (2010) show the application of ABC algorithm in software testing. Rao et al. (2008) used the ABC algorithm to solve network reconfiguration problem in a radial distribution system. Furthermore they compared, the results obtained by ABC algorithm with the other methods and found that ABC outperformed in terms of quality of the objective value and computational efficiency. Bendeş and Özkan (2008) applied the ABC algorithm for solving direct linear transformation (DLT) and results are compared against those of the DE algorithm. DLT is a camera calibration method which establishes a relation between 3D object coordinate and 2D image plane linearly. Linh and Anh (2010) used the ABC for determining the sectionalising switch to be operated in order to solve the distribution system loss minimisation problem. Karaboga (2009) applied the ABC algorithm in the signal processing area for designing digital IIR filters. Horng and Jiang (2011) solved the image vector quantisation problem via honey bee mating optimisation strategy. The results were compared with the other three methods that are Linde-Buzo-Gray (LBG), PSO-LBG and quantum PSO LBG algorithms. In order to improve the global search ability of ABC, Xing et al. (2007) worked in the control mechanism of local optimal solution. Ma et al. (2011) show the synthetic aperture radar (SAR) image segmentation based on ABC algorithm. Pawar et al. (2008) applied the ABC algorithm in mechanical engineering problems, including multi-objective optimisation of electro-chemical machining process parameters, optimisation of process parameters of the abrasive flow machining process and the milling process.

There are two fundamental processes which drive the swarm to update in ABC: the variation process, which enables exploring different areas of the search space, and the selection process, which ensures the exploitation of the previous experience. However, it has been shown that ABC may occasionally stop proceeding toward the global optimum even though the population has not converged to a local optimum (Karaboga and Akay, 2009). Therefore, to maintain the proper balance between exploration and exploitation behaviour of ABC, a new control parameter called cognitive learning factor (CLF) is introduced in ABC and the range of ABC control parameter ϕ_{ij} (used in ABC's food position update equation) is modified. ABC with these modifications is named as balanced artificial bee colony ($BABC$). In terminology of social science, Cognitive Learning is about enabling people to learn by using their reason, intuition and perception. This technique is often used to change people's behaviour. The same phenomenon is also applied in $BABC$. In $BABC$, a weight factor (CLF) is associated with the individual's experience in the position update process of employed bees phase and onlooker bees phase of ABC. Furthermore, the range of control parameter ϕ_{ij} is also varied from [–1, 1] to [–0.25, 0.25]. By varying this weight (CLF) and range of ϕ_{ij}, the exploration and exploitation capabilities of ABC or any modified version of ABC may be modified.

Rest of the paper is organised as follows: Section 2 describes brief overview of basic ABC algorithm. In Section 3, some basic Improvements on ABC algorithm are briefly reviewed. $BABC$ algorithm is proposed and tested in Section 4. In Section 5, $BABC$ concept is applied to a recent variant of ABC called best-so-far in ABC and a comparative study has been carried out. Finally, in Section 6, paper is concluded.

2 Brief overview of ABC algorithm

Swarm-based optimisation algorithms find solution by collaborative trial and error. Peer to peer learning behaviour of social colonies is the main driving force behind the development of many efficient swarm-based optimisation algorithms. ABC optimisation algorithm is a recent addition in this category. Like any other population-based optimisation algorithm, ABC consists of a population of potential solutions. With reference to ABC, the potential solutions are food sources of honey bees. The fitness is determined in terms of the quality (nectar amount) of the food source. The total number of bees in the colony are divided into three groups: onlooker bees, employed bees and scout bees. Number of employed bees or onlooker bees are equal to the food sources. Employed bees are associated with food sources while onlooker bees are those bees that stay in the hive and use the information gathered from employed bees to decide the food source. Scout bee searches the new food sources randomly.

Similar to the other population-based algorithms, ABC is an iterative process. ABC process requires cycles of four phases: initialisation phase, employed bees phase, onlooker bees phase and scout bee phase. Each of the phase is explained as follows.

2.1 Initialisation of the population

Initially, ABC generates a uniformly distributed initial population of SN solutions where each solution $x_i(i = 1, 2, ..., SN)$ is a D-dimensional vector. Here D is the number of variables in the optimisation problem and x_i represent the i^{th} food source in the population. Each food source is generated as follows:.

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}) \tag{1}$$

where x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction and $rand[0, 1]$ is a uniformly distributed random number in the range [0, 1]

2.2 Employed bee phase

In employed bee phase, employed bees modify the current solution based on the information of individual experience and the fitness value of the new solution (nectar amount). If the fitness value of the new source is higher than that of the old source, the bee updates her position with the new one and discards the old one. The position update equation for i^{th} candidate in this phase is

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., D\}$ are randomly chosen indices. k must be different from i. ϕ_{ij} is a random number between [–1, 1].

2.3 Onlooker bees phase

After completion of the employed bees phase, the onlooker bees phase starts. In onlooker bees phase, all the employed bees share the new fitness information (nectar) of the new solutions (food sources) and their position information with the onlooker bees in the hive. Onlooker bees analyse the available information and select a solution with a probability, p_i, related to its fitness. The probability p_i may be calculated using following expression (there may be some other but must be a function of fitness):

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{3}$$

where fit_i is the fitness value of the solution i. As in the case of the employed bee, she produces a modification on the position in her memory and checks the fitness of the candidate source. If the fitness is higher than that of the previous one, the bee memorises the new position and forgets the old one.

2.4 Scout bees phase

If the position of a food source is not updated up to predetermined number of cycles, then the food source is assumed to be abandoned and scout bees phase starts. In this phase the bee associated with the abandoned food source becomes scout bee and the food source is replaced by a randomly chosen food source within the search space. In ABC, predetermined number of cycles is a crucial control parameter which is called $limit$ for abandonment.

Assume that the abandoned source is x_i. The scout bee replaces this food source by a randomly chosen food source which is generated as follows

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}), \text{ for } j \in \{1, 2, ..., D\} \tag{4}$$

where x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction.

2.5 Main steps of the ABC algorithm

It is clear from the above discussion that there are three control parameters in ABC search process: The number of food sources SN (equal to number of onlooker or employed bees), the value of $limit$ and the maximum number of cycles MCN.

In the ABC algorithm, the exploitation process is carried out by onlooker and employed bees and exploration process is carried out by scout bees in the search space. The pseudo-code of the ABC algorithm is shown in Algorithm 1 (Karaboga and Akay, 2009):

Algorithm 1 ABC algorithm

```
Initialise the population of solutions, x_i (i = 1, 2, ...; SN) by using (1);
cycle = 1;
while cycle <> MCN do
   Produce new solutions v_i for the employed bees using (2) and evaluate them;
   Apply the greedy selection process for the employed bees;
   Calculate the probability values p_i for the solutions x_i;
   Produce the new solutions v_i for the onlookers from the solutions x_i selected depending
   on p_i using (3) and evaluate them;
   Apply the greedy selection process for the onlookers;
   Determine the abandoned solution for the scout, if exists, and replace it with a new
   randomly produced solution x_i using (4);
   Memorise the best solution achieved so far;
   cycle = cycle + 1;
end while
```

3 Brief review on basic improvements in ABC algorithm

Often real world provides some complex optimisation problems that can not be easily dealt with available mathematical optimisation methods. If the user is not very conscious about the exact solution of the problem in hand then intelligence emerged from social behaviour of social colony members may be used to solve these kind of problems.

Honey bees are in the category of social insects. The foraging behaviour of honey bees produces an intelligent social behaviour, called as swarm intelligence. This swarm intelligence is simulated and an intelligent search algorithm namely, ABC algorithm is established by Karaboga (2005). Since its inception, a lot of research has been carried out to make ABC more and more efficient and to apply ABC for different types of problems.

In order to get rid of the drawbacks of basic ABC, researchers have improved ABC in many ways. The potentials where ABC can be improved may be broadly classified into three categories:

- fine tuning of ABC control parameters $SN, \phi_{ij}, limit$
- hybridisation of ABC with other population-based probabilistic or deterministic algorithms
- introducing new control parameters.

This paper concentrates on the third area of ABC research, i.e., the paper introduces a new control parameter, namely, CLF in ABC process.

Karaboga (2005) have observed that the value of ϕ_{ij} should be in the range of [–1, 1]. The value of $limit$ should be $SN \times D$, where, SN is the number of solutions and D is the dimension of the problem.

Gao and Liu (2011) proposed an improved solution search equation in ABC, which is based on the fact that bee searches only around the best solution of the previous iteration to improve the exploitation. Banharnsakun et al. (2010b) introduced a new variant of ABC namely the best-so-far selection in ABC algorithm. To enhance the exploitation and exploration processes, they propose to make three major changes by introducing the best-so-far method, an adjustable search radius, and an objective-value-based comparison method in DE.

Haijun and Qingxian (2008) proposed a modification in the initialisation scheme by making the initial group symmetrical, and the Boltzmann selection mechanism was employed instead of roulette wheel selection for improving the convergence ability of the ABC algorithm.

In order to maximise the exploitation capacity of the onlooker stage, Tsai et al. (2009) introduced the Newtonian law of universal gravitation in the onlooker phase of the basic ABC algorithm in which onlookers are selected based on a roulette wheel (interactive ABC, IABC). Baykasoglu et al. (2007) incorporated the ABC algorithm with shift neighbourhood searches and greedy randomised adaptive search heuristic and applied it to the generalised assignment problem.

Furthermore, a modified versions of the ABC algorithm are introduced and applied for efficiently solving real-parameter optimisation problems by Akay and Karaboga (2012). In the proposed work, effects of the perturbation rate that controls the frequency of parameter change, the scaling factor (step size) that determines the magnitude of change in parameters while producing a neighbouring solution, and the $limit$ parameter on the performance of the ABC algorithm are investigated on real-parameter optimisation.

4 Balanced ABC

4.1 A few drawbacks of ABC

The inherent drawback with most of the population-based stochastic algorithms is premature convergence. ABC is not an exception. Any population-based algorithm is regarded as an efficient algorithm if it is fast in convergence and able to explore the maximum area of the search space. In other words, if a population-based algorithm is capable of balancing between exploration and exploitation of the search space, then the algorithm is regarded as an efficient algorithm. From this point of view, basic ABC is not an efficient algorithm (Karaboga and Akay, 2009). Karaboga and Akay (2009) compared the different variants of ABC for global optimisation and found that ABC shows a poor performance and remains inefficient in exploring the search space. The problems of premature convergence and stagnation is a matter of serious consideration for designing a comparatively efficient[1] ABC algorithm.

4.2 Motivation for $BABC$

This section proposes a new control parameter, CLF as well as modified range of ϕ in ABC algorithm.

4.2.1 Effect of CLF and ϕ_{ij} in ABC

Exploration of the whole search space and exploitation of the near optimal solution region may be balanced by maintaining the diversity in early and later iterations for any random number-based search algorithm. Position update equation (2) in employed bees phase and onlooker bees phase in ABC may be seen in the following way:

$$v_{ij} = A \times x_{ij} + B \times (x_{ij} - x_{kj}) \tag{5}$$

i.e., modified position v_{ij} is the weighted sum of the food source position x_{ij} and the difference $(x_{ij} - x_{kj})$ of two food source positions. Here, A is the weight to target food source and B is the weight to the difference of random food source and target food source. In basic ABC, A is set to be 1, while B is a uniformly distributed real random number (ϕ_{ij}) in the range [–1, 1]. Studies have been carried out with varying (ϕ_{ij}) for better exploration and exploitation mechanism (Akay and Karaboga, 2012). In this paper, experiments are performed over 24 benchmark problems to find an optimal strategy to set the weight A and B. Here, A named as CLF and denoted by 'C' (for this paper) and B is represented by ϕ_{ij} as usual. CLF is the weight to individual's current position or in other words, this is the weight to self confidence and therefore, it is named so.

Hence, the modified position update equation in employed bees phase and onlooker bees phase of cognitive learning ABC becomes:

$$v_{ij} = C \times x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{6}$$

Symbols have their usual meanings. In the proposed strategy, value of C is linearly increased (0.1 to 1) while the range of control parameter ϕ_{ij} is linearly decreased ([–1, 1] to [–0.25, 0.25]) by iterations. It is clear from equation (6) that low value of C and high value of ϕ_{ij}, increase the exploration capability as the weight for current food position is low whereas weight for variation is high. Furthermore, high value of C and low value of ϕ_{ij}, increase exploitation capability as in this case, weight to current food position is high whereas weight to variation of food source position is low. Therefore, by linearly increasing (0.1 to 1) CLF and linearly decreasing ϕ_{ij} ([–1, 1] to [–0.25, 0.25]), diversity will be relatively high in early iterations and will keep on reducing in successive iterations while convergence rate behaviour is expected to be opposite of the diversity. So, it is expected that these modifications should improve the results.

The $BABC$ algorithm is similar to the basic ABC algorithm except the position update process of employed bees as well as onlooker bees. The pseudo-code of the $BABC$ algorithm is shown in Algorithm 2.

Algorithm 2 Balanced artificial bee colony

Initialise the population of solutions, $x_i(i = 1, 2, ...; SN)$ by using (1);
cycle = 1;
while cycle <> MCN **do**
  Produce new solutions v_i for the employed bees using following equation and evaluate them;

$$v_{ij} = C \times x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

  Apply the greedy selection process between v_i and x_i for the employed bees;
  Calculate the probability values P_i for the solutions x_i;
  Produce the new solutions v_i for the onlookers from the solutions x_i selected depending on p_i using following equation and evaluate them;

$$v_{ij} = C \times x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

  Apply the greedy selection process between v_i and x_i for the onlookers;
  Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution x_i using (4);
  Memorise the best solution achieved so far;
  cycle = cycle + 1;
end while

4.3 Control parameters in $BABC$

As stated by Karaboga (2005) and Akay and Karaboga (2012), ABC performance is very sensitive to the choice of ϕ_{ij} and $limit$. Some settings of control parameters are suggested by Karaboga (2005):

- $\phi_{ij} = rand[-1, 1]$
- $limit$ should be $SN \times D$
- SN should be equal to the number of employed bees or onlooker bees.

$BABC$ introduces one new parameter 'C' named CLF, therefore, now there are four controlling parameters $(\phi_{ij}, limit, SN, C)$ in $BABC$. CLF C is an important parameter in $BABC$ as it is responsible for better balance between the exploration and exploitation capabilities of the algorithm. C is linearly increased from 0.1 to 1 as follows:

$$C_{i+1} = C_i + \frac{(1 - 0.1)}{N}$$

where N is the total number of iterations. The range of ϕ_{ij} also modified and vary from [–1, 1] to [–0.25, 0.25]. This scheme is expected to produce more diversity and slow convergence in initial iterations and fast convergence to a good solution in later iterations. In the next two sections $BABC$ is tested over 24 benchmark problems.

Table 1 Test problems

S. no.	*Test problem*	*Objective function*	*Search space*
1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	[–5.12 5.12]
2	De Jong f4	$f(x) = \sum_{i=1}^{n} i.(x_i)^4$	[–5.12 5.12]
3	Griewank	$f(x) = 1 + \frac{1}{4,000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$	[–600 600]
4	Rosenbrock	$f(x) = \sum_{i=1}^{n}(100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$	[–30 30]
5	Rastrigin	$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	[–5.12 5.12]
6	Ackley	$f(x) = -20 + e + exp(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n} x_i{}^3})$ $- \ exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\ (2\pi.x_i)x_i\right)$	[–1 1]
7	DropWave	$f(x) = -\frac{1+cos(\ ^{12}\sqrt{\sum_{i=1}^{n} x_i{}^2})}{\frac{1}{2}\sum_{i=1}^{n} x_i{}^2+2}$	[–5.12 5.12]
8	Alpine	$f(x) = \sum_{i=1}^{n}\lvert x_i \sin\ x_i + 0.1x_i\rvert$	[–10 10]
9	Michalewicz	$f(x) = -\sum_{i=1}^{n}\sin\ x_i(\sin\ (\frac{i.x_i{}^2}{\pi}))$	[0π]
10	Cosine mixture	$f(x) = \sum_{i=1}^{n} x_i{}^2$ $-\ 0.1(\sum_{i=1}^{n}\cos\ 5.\pi.x_i) + 0.1n$	[–1 1]
11	Exponential	$f(x) = -(exp(-0.5\sum_{i=1}^{n} x_i{}^2)) + 1$	[–1 1]
12	Zakharov	$f(x) = \sum_{i=1}^{n} x_i{}^2 + (\sum_{i=1}^{n}\frac{ix_i}{2})^2$ $+\ \left(\sum_{i=1}^{n}\frac{ix_1}{2}\right)^4$	[–5.12 5.12]
13	Cigar	$f(x) = x_0{}^2 + 100,000\sum_{i=1}^{n} x_i{}^2$	[–10 10]
14	brown3	$f(x) = \sum_{i=1}^{n-1}(x_i{}^{2x_{i+1}{}^2+1} + x_{i+1}{}^{2x_i{}^2+1})$	[–1 4]
15	Schwefel	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	[–10 10]
16	Salomon problem (SAL)	$f(x) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right)$ $+\ 0.1\left(\sqrt{\sum_{i=1}^{n} x_i^2}\right)$	[–100 100]
17	Axis parallel hyper-ellipsoid function	$f(x) = \sum_{i=1}^{n} i.x_i^2$	[–5.12 5.12]
18	Pathological function	$f(x)$ $= \sum_{i=1}^{n-1}\left(0.5 + \frac{sin^2\sqrt{(100x_i^2+x_{i+1}^2)}-0.5}{1+0.001(x_i^2-2x_ix_{i+1}+x_{i+1}^2)^2}\right)$	[–100 100]
19	Sum of different powers	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	[–1 1]
20	Step function	$f(x) = \sum_{i=1}^{n}(\lfloor x_i + 0.5\rfloor)^2$	[–100 100]
21	Quartic function, i.e., noise	$f(x) = \sum_{i=1}^{n} i.x_i^4 + \text{random}[0,1)$	[–1.28 1.28]
22	Inverted cosine wave function	$f(x)$ $= -\sum_{i=1}^{n-1}\left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})}{8}\right)\times \mathrm{I}\right)$ where $\mathrm{I} = cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_ix_{i+1}}\right)$	[–5 5]
23	Neumaier 3 problem (NF3)	$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_ix_{i-1}$	$[D^2 D^2]$
24	Rotated hyper-ellipsoid function	$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{i} x_j^2$	[–65.536 65.536]

5 Experimental results and discussion

5.1 Test problems under consideration

In order to see the effect of CLF on ABC, 24 scalable (the number of decision variables may be varied as per user's choice) test problems of optimisation are selected (listed in Table 1). These are continuous optimisation problems and have different degrees of complexity and multimodality. For this study, number of decision variables is set to 30.

5.2 Experimental setting

To test ABC and ABC variants over test problems, following experimental setting is adopted:

- colony size SN=100
- $\phi_{ij} = rand[-1, 1]$ and for $BABC$ range of ϕ_{ij} is linearly decreased from $[-1, 1]$ to [–0.25, 0.25]
- number of food sources $SN/2$
- $limit$ = 1,500
- the stopping criteria is either maximum number of iterations (which is set to be 1,000) is reached or the objective function value ≤ 0.0000001
- the number of simulations/run = 100
- the number of decision variables in scalable test problems D = 30.

5.3 Comparison of $BABC$ with ABC

Numerical results with experimental setting of Section 5.2 are given in Table 2. In Table 2, success rate (SR) (a simulation is said to be successful if the objective function value is ≤ 0.0000001 in iterations up to 1,000), mean objective function value $(MOFV)$, average function evaluations (AFE), and standard deviation (SD) are reported. Table 2 shows that most of the time inclusion of CLF and fine tuning of ϕ_{ij} in ABC improves the reliability, efficiency and accuracy.

Further, Figure 1 shows the convergence characteristics in terms of the error of the median run of each algorithm for functions on which both the algorithms achieved 100% success rate within the specified maximum function evaluations (to carry out fair comparison of convergence rate). It can be observed that the convergence of $BABC$ is relatively better than the basic ABC.

Some more intensive statistical analyses based on t test and boxplots have been carried out for results of the basic ABC and $BABC$.

Table 2 Comparison of the results of test problems

Test problem	*Algorithm*	*MOFV*	*SD*	*AFE*	*SR*
Sphere	ABC	8.02E-08	2.01E-08	53,396	100
	BABC	7.45E-08	2.15E-08	22,469	100
De Jong	ABC	5.58E-08	3.04E-08	22,540	100
	BABC	4.74E-08	3.18E-08	9,934	100
Griewank	ABC	9.30E-08	1.10E-07	85,687	90
	BABC	6.49E-08	2.67E-08	33,203	100
Rosenbrock	ABC	7.53E-01	7.22E-01	100,000	0
	BABC	2.54E+01	1.27E+00	100,000	0
Rastrigin	ABC	3.05E-02	1.70E-01	94,389	67
	BABC	6.51E-08	2.64E-08	32,728	100
Ackley	ABC	4.46E-06	2.54E-06	100,000	0
	BABC	8.10E-08	1.85E-08	49,182	100
DropWave	ABC	3.06E-08	2.75E-08	379	100
	BABC	3.29E-08	2.83E-08	451	100
Alpine	ABC	1.62E-04	1.30E-04	100,000	0
	BABC	8.37E-08	1.77E-08	53,531	100
Michalewicz	ABC	7.01E-07	8.06E-07	93,437.4	17
	BABC	5.78E-08	3.94E-08	63,528.12	96
Cosine mixture	ABC	7.55E-08	2.23E-08	55,176	100
	BABC	7.07E-08	2.54E-08	22,662	100
Exponential	ABC	7.57E-08	2.28E-08	44,511	100
	BABC	7.07E-08	2.27E-08	19,288	100
Zakharov	ABC	1.20E+02	1.59E+01	100,000	0
	BABC	1.03E+02	1.47E+01	100,000	0
Cigar	ABC	7.51E-08	2.30E-08	83,412	100
	BABC	7.75E-08	2.20E-08	35,993	100
brown3	ABC	7.61E-08	2.14E-08	55,410	100
	BABC	7.00E-08	2.16E-08	22,698	100
Schwefel	ABC	2.31E-07	1.29E-07	99,903	10
	BABC	8.69E-08	1.53E-08	45,473	100
Salomon problem	ABC	1.59E+00	2.02E-01	100,000	0
	BABC	9.02E-01	1.27E-01	100,000.08	0
Axis parallel	ABC	7.89E-08	1.98E-08	58,400	100
hyper-ellipsoid function	BABC	7.68E-08	2.22E-08	25,099	100
Pathological function	ABC	4.15E+00	3.25E-01	100,000	0
	BABC	1.25E+00	4.70E-01	100,000	0
Sum of different powers	ABC	5.85E-08	2.73E-08	49,387	100
	BABC	5.06E-08	2.77E-08	21,132	100
Step function	ABC	0.00E+00	0.00E+00	20,837	100
	BABC	0.00E+00	0.00E+00	8,494	100
Quartic function	ABC	1.23E+01	5.57E-01	100,003.92	0
	BABC	9.94E+00	4.18E-01	100,008.07	0
Inverted cosine	ABC	5.17E-08	2.97E-08	6,121	100
wave function	BABC	5.37E-08	3.15E-08	7,745	100
Neumaier 3 problem	ABC	–4.22E+01	2.47E+02	61,774	95
	BABC	–4.18E+01	6.00E+01	22,917	100
Rotated hyper-ellipsoid function	ABC	7.78E-08	2.23E-08	71,014	100
	BABC	7.66E-08	2.00E-08	30,269	100

Figure 1 Convergence characteristics of ABC and $BABC$ for functions, (a) sphere (b) De Jong's (c) dropwave (d) cosine mixture (e) exponential (f) cigar (g) brown3 (h) axis parallel hyperellipsoid (i) sum of different powers (j) step function (k) inverted cosine wave function (l) rotated hyper-ellipsoid (see online version for colours)

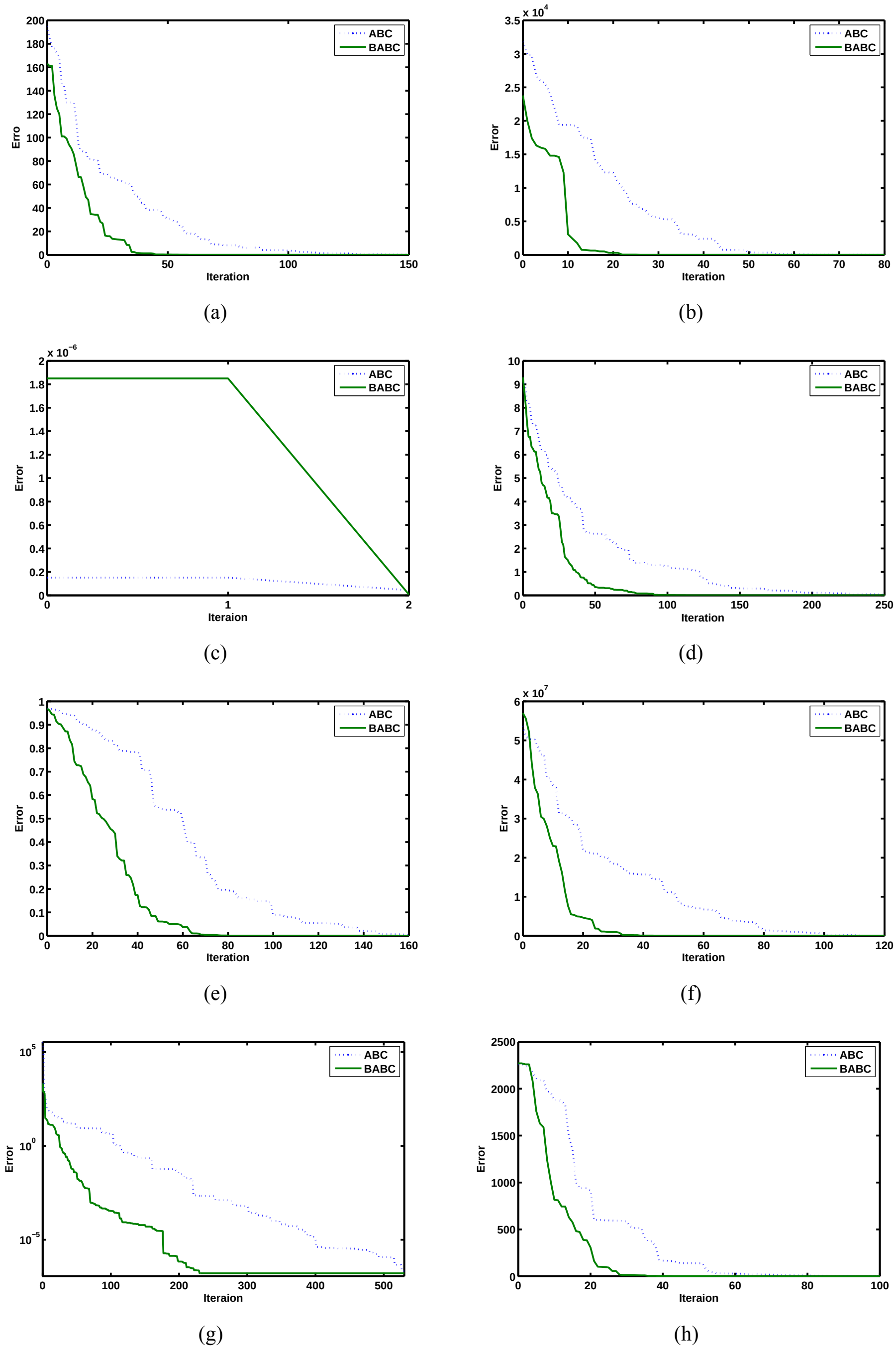
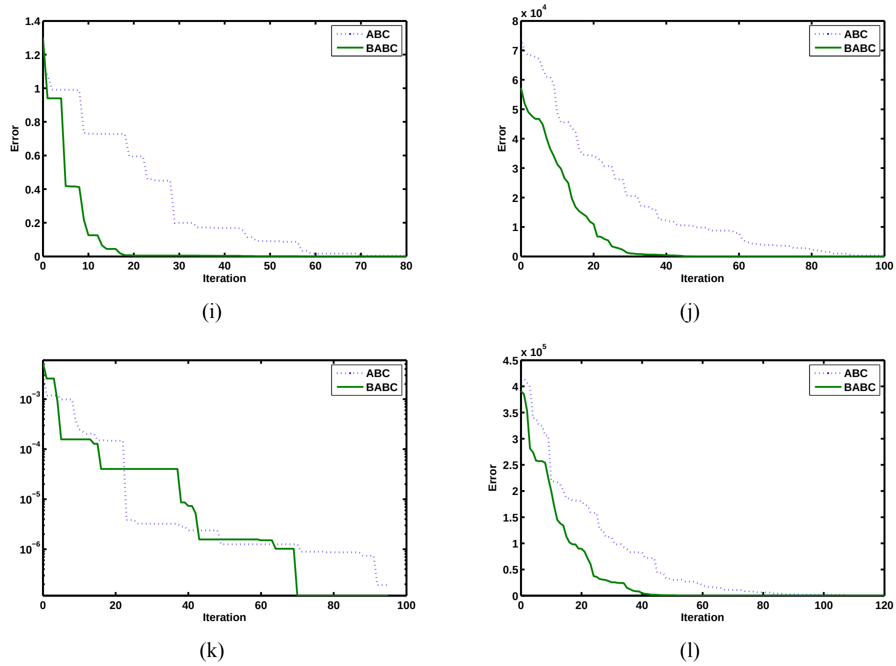


Figure 1 Convergence characteristics of ABC and $BABC$ for functions, (a) sphere (b) De Jong's (c) dropwave (d) cosine mixture (e) exponential (f) cigar (g) brown3 (h) axis parallel hyperellipsoid (i) sum of different powers (j) step function (k) inverted cosine wave function (l) rotated hyper-ellipsoid (continued) (see online version for colours)



5.3.1 Statistical analysis

In order to extract the best strategy of setting range of control parameter ϕ and CLF in ABC, a comparative analysis is done for ABC and $BABC$. Statistical comparisons have been carried out using t-test and boxplots.

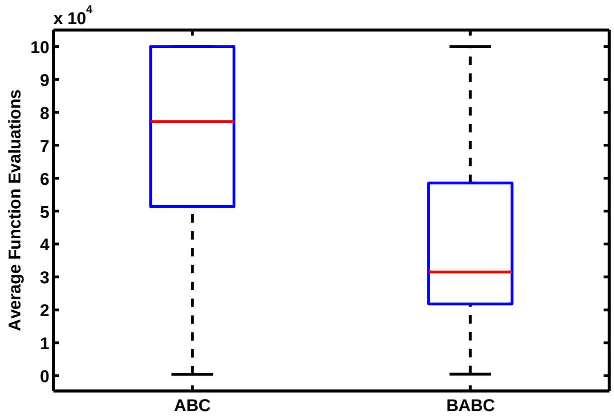
The t-test is quite popular among researchers in the field of evolutionary computation. In this pape,r students t-test is applied according to the description given in Croarkin and Tobias (2010) for a confidence level of 0.95. Table 3 shows the results of the t-test for the null hypothesis that there is no difference in the mean number of function evaluations of 100 runs using the basic ABC and $BABC$. Note that here '+' indicates the significant difference (or the null hypothesis is rejected) at a 0.05 level of significance, '–' implies that there is no significant difference while '=' indicates that comparison is not possible. In Table 3, $BABC$ is compared with the ABC. It is observed that significant differences observed in 23 comparisons out of 24 comparisons. Therefore, it can be concluded that the results of $BABC$ is significantly better than the basic ABC.

For the purpose of comparison in terms of consolidated performance, boxplot analysis is carried out. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool (Williamson et al., 1989). The boxplots for ABC and $BABC$ are shown in Figure 2. It is clear from this figure that $BABC$ is better than the basic ABC as Interquartile Range and Median are low for $BABC$.

Table 3 Results of the student's t test

S. no.	*Test problem*	*ABC*	*S. no.*	*Test problem*	*ABC*
1	Sphere	+	13	Cigar	+
2	De Jong f4	+	14	brown3	+
3	Griewank	+	15	Schwefel	+
4	Rosenbrock	+	16	Salomon problem (SAL)	+
5	Rastrigin	+	17	Axis parallel hyper-ellipsoid function	+
6	Ackley	+	18	Pathological function	+
7	DropWave	+	19	Sum of different powers	+
8	Alpine	+	20	Step function	+
9	Michalewicz	–	21	Quartic function, i.e., noise	+
10	Cosine mixture	+	22	Inverted cosine wave function	+
11	Exponential	+	23	Neumaier 3 problem (NF3)	+
12	Zakharov	+	24	Rotated hyper-ellipsoid function	+

Figure 2 Boxplot graph for average function evaluation (see online version for colours)



5.4 *BSFABC and GABC with same modifications*

It is obvious from Section 5.3 that the $BABC$ performs better with the linearly increasing CLF and linearly decreasing range of ϕ. The experimental findings support our theoretical suggestions that these modifications in ABC should produce relatively better results.

Further, it will be interesting to investigate that whether linearly increasing CLF and linearly decreasing range of ϕ improve the performance of some modified versions of ABC. In this paper, these modifications are tested with best-so-far selection in ABC algorithm $(BSFABC)$ (Banharnsakun et al., 2010b) and gbest-guided ABC $(GABC)$ algorithm (Zhu and Kwong, 2010). The $BSFABC$ and $GABC$ algorithm, with CLF and modified range of ϕ are denoted as balanced $BSFABC$ ($BBSFABC$) and balanced $GABC$ ($BGABC$) respectively. The experimental results are shown in Tables 4 and 5 respectively. The experiments are carried out on the same set of benchmark optimisation functions on which authors of $BSFABC$ and $GABC$ performed the experiments. It can be observed that except for Rosenbrock problem, modifications improve the performance of both the algorithms.

Table 4 Experimental results of $BSFABC$ and $BBSFABC$

Test problem	*Search space*	*Dimension*	*GABC*				*BGABC*			
			MOFV	*SD*	*AFE*	*SR*	*MOFV*	*SD*	*AFE*	*SR*
Sphere	[–5.12,5.12]	10	2.88E–07	0.455636	15,598	100	4.57E–08	0.00715454	4,522	100
		30	2.61E–07	1.45216	26,882	100	4.81E–08	0.00911847	4,782	100
		50	3.15E–07	2.21658	48,150	100	1.20E–08	0.0157341	4,990	100
Rastrigin	[–5.12, 5.12]	10	2.70E–07	11.7469	51,764	100	2.07E–08	0.0668531	5,016	100
		30	3.16E–07	39.9666	93,754	100	1.53E–08	0.140971	5,120	100
		50	3.05E–07	68.2341	103,452	100	9.89E–09	0.114436	5,068	100
Rosenbrock	[–30, 30]	10	4.50E–07	9889.7	667,548	100	7.53906	2.3824	2.60E+06	0
		30	1.42E–06	76,465.4	2.26E+06	92	21.8315	9.04951	2.60E+06	0
		50	0.0447764	91,909.2	2.60E+06	3	33.8833	15.9404	2.60E+06	0
Griewank	[–600, 600]	10	2.56E–07	2.74128	31,094	100	5.46E–08	0.155223	5,016	100
		30	2.87E–07	7.96189	51,790	100	2.04E–08	0.303384	5,042	100
		50	2.61E–07	12.0349	70,562	100	2.66E–08	0.341456	5,198	100
Ackley	[–30, 30]	10	8.09E–07	10.5242	927,548	99	3.55E–08	5.83827	21,032	100
		30	7.13E–07	11.9812	881,424	100	7.82E–08	6.91307	15,442	100
		50	7.75E–07	14.7213	957,578	99	1.67E–08	6.89608	14,896	100
Schaffer	[–100, 100]	2	3.58E–07	0.143907	105,454	100	1.64E–07	0.0553836	12,634	100

Table 5 Experimental results of *GABC* and *BGABC*

Test problem	*Search space*	*Dimension*	*GABC*				*BGABC*			
			MOFV	*SD*	*AFE*	*SR*	*MOFV*	*SD*	*AFE*	*SR*
Sphere	[−5.12, 5.12]	25	8.06E−08	1.60E−08	20,583	100	8.09E−08	1.89E−08	25,792	100
		50	8.78E−08	1.21E−08	44,995	100	8.11E−08	1.82E−08	47,012	100
		75	8.50E−08	1.34E−08	70,501	100	8.39E−08	1.53E−08	65,232	100
Rastrigin	[−5.12, 5.12]	25	7.17E−08	2.42E−08	40,036	100	5.88E−08	2.76E−08	41,772	100
		50	8.34E−05	4.24E−04	94,463	64	7.27E−08	2.49E−08	76,086	100
		75	1.82E+00	1.14E+00	100,000	0	2.06E−03	9.03E−03	100,000	0
Rosenbrock	[−30, 30]	25	3.92E+00	1.26E+01	100,000	0	2.52E+01	1.45E+01	100,000	0
		50	7.53E+00	1.43E+01	100,000	0	6.92E+01	3.14E+01	100,000	0
		75	3.83E+01	3.86E+01	100,000	0	1.20E+02	4.00E+01	100,000	0
Griewank	[−600, 600]	25	1.73E−04	1.22E−03	35,596	98	1.23E−04	1.34E−03	30,957	100
		50	3.95E−04	1.96E−03	63,370	95	7.60E−05	7.53E−04	61,240	98
		75	3.94E−04	3.26E−03	90,615	90	2.93E−04	2.10E−03	81,207	97
Ackley	[−30, 30]	25	8.91E−08	1.13E−08	41,772	100	8.72E−08	1.34E−08	43,402	100
		50	9.21E−08	7.80E−09	88,940	100	9.18E−08	7.98E−09	83,457	100
		75	3.72E−05	8.94E−06	100,000	0	1.02E−05	2.89E−06	100,000	0
Schaffer	[−100, 100]	2	1.00E−04	9.67E−04	23,483.21	95	9.72E−05	9.67E−04	21,016.74	98
		3	2.85E−03	4.35E−03	86,144.1	30	2.73E−03	4.36E−03	78,181.19	47
		4	6.36E−03	4.58E−03	98,778.91	3	7.49E−03	4.08E−03	99,155.55	3

More intensive comparative analyses using statistical tools; t-test and boxplots between $BSFABC$ and $BBSFABC$, $GABC$ and $BGABC$ has also been carried out. Tables 6 and 7 show the results of the t-test respectively. It can be observed from Tables 6 and 7 that in 15 comparisons out of 16 comparisons $BBSFABC$ outperforms than the $BSFABC$ and in 11 comparisons out of 18 comparisons $BGABC$ outperforms than the $GABC$. Therefore, it can be concluded that results of $BBSFABC$ and $BGABC$ are significantly better than the $BSFABC$ and $GABC$ respectively.

Table 6 Results of the student's t test of $BSFABC$ with $BBSFABC$

Test problem	*Dimension*	*BSFABC*	*Test problem*	*Dimension*	*BSFABC*
Sphere	10	+	Griewank	10	+
	30	+		30	+
	50	+		50	+
Rastrigin	10	+	Ackley	10	+
	30	+		30	+
	50	+		50	+
Rosenbrock	10	+	Schaffer	2	+
	30	+			
	50	–			

Table 7 Results of the student's t test of $GABC$ with $BGABC$

Test problem	*Dimension*	*GABC*	*Test problem*	*Dimension*	*GABC*
Sphere	25	–	Griewank	25	+
	50	–		50	+
	75	+		75	+
Rastrigin	25	–	Ackley	25	+
	50	+		50	+
	75	+		75	+
Rosenbrock	25	–	Schaffer	2	+
	50	–		3	+
	75	–		4	–

For the purpose of comparison in terms of consolidated performance, boxplot analysis is carried out. Boxplots based on the average function evaluations of $BSFABC$-$BBSFABC$ and $GABC$-$BGABC$ are shown in Figures 3 and 4 respectively.

It is observed by boxplot analysis that interquartile Range and Median of $BBSFABC$ is significantly less than that of $BSFABC$ and Median of $BGABC$ is less than that of $GABC$. Hence, the performance of $BSFABC$ and $GABC$ is significantly improved after incorporating CLF with the modify range of ϕ. Through t-test and boxplots, we can say that the effect of the modifications is significant on the performance of $BSFABC$ and $GABC$ algorithms.

Figure 3 Boxplot graph for average function evaluation of $BSFABC$ and $BBSFABC$ (see online version for colours)

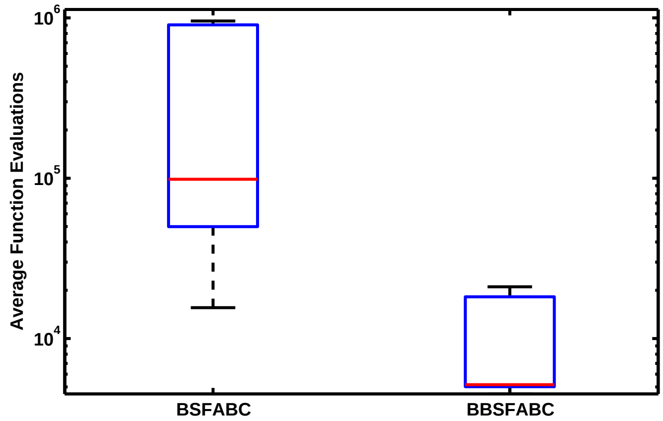
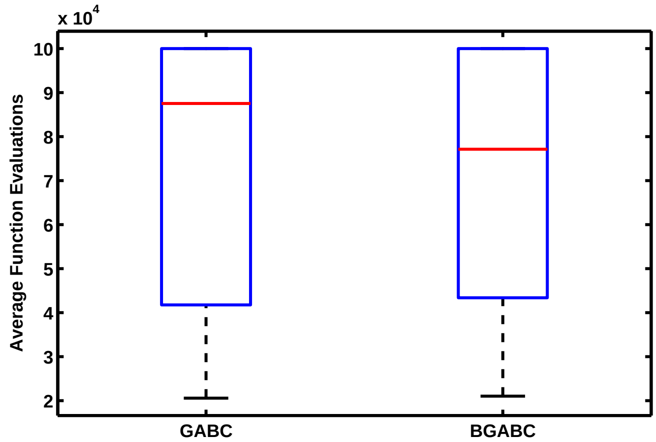


Figure 4 Boxplot graph foraverage function evaluation of $GABC$ and $BGABC$ (see online version for colours)



6 Conclusions

In this paper, basic ABC algorithm is improved by introducing a new control parameter (CLF) in ABC search procedure and linearly decreasing range of control parameter ϕ of position update equation from [–1, 1] to [–0.25, 0.25]. With the help of experiments over test problems, it has been shown that the reliability (due to success rate), efficiency (due to average number of function evaluations) and accuracy (due to mean objective function value) of basic as well as modified versions of ABC, namely $BSFABC$ and $GABC$ algorithms with these modifications are higher than that of its original versions. The modified ABC so obtained is named as $BABC$.

Based on this study, it is concluded that $BABC$ is a better candidate in the field of nature inspired algorithms for function optimisation. The future scope of this work is the implementation of these modifications to other biologically inspired algorithms.

Acknowledgements

Harish Sharma acknowledges ABV-Indian Institute of Information Technology and Management Gwalior, India for providing research grant to carry out this work.

References

Akay, B. and Karaboga, D. (2008) 'Effect of region scaling on the initialization of particle swarm optimization differential evolution and artificial bee colony algorithms on multimodal high dimensional problems', in *International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining (Kayseri, TURKEY)*, June 19–23.

Akay, B. and Karaboga, D. (2012) 'A modified ABC algorithm for real-parameter optimization', *Information Sciences*, June, Vol. 192, pp.120–142.

Akay, B., Karaboga, D. and Ozturk, C. (2008) 'Training neural networks with abc optimization algorithm on medical pattern classification', *International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining (Kayseri, TURKEY)*, June 19–23.

Banharnsakun, A., Achalakul, T. and Sirinaovakul, B. (2010a) 'Artificial bee colony algorithm on distributed environments', in *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pp.13–18, IEEE.

Banharnsakun, A., Achalakul, T. and Sirinaovakul, B. (2010b) 'The best-so-far selection in artificial bee colony algorithm', *Applied Soft Computing*, Vol. 11, No. 2, pp.2888–2901.

Baykasoglu, A., Ozbakir, L. and Tapkan, P. (2007) 'Artificial bee colony algorithm and its application to generalized assignment problem', *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, December, pp.113–144, Itech Education and Publishing, Vienna, Austria, ISBN 978-3-902613-09-7, 532pp.

Bendeş, E. and Özkan, C. (2008) 'Direk lineer trasformasyon yönteminde yapay zeka tekniklerinin uygulanması', September 13–15.

Croarkin, C. and Tobias, P. (2010) 'Nist/sematech e-handbook of statistical methods', Retrieved March, Vol. 1.

Dahiya, S.S., Chhabra, J.K. and Kumar, S. (2010) 'Application of artificial bee colony algorithm to software testing', in *21st Australian Software Engineering Conference*, pp.149–154, IEEE.

Dorigo, M. and Stutzle, T. (2004) *Ant Colony Optimization*, The MIT Press, Cambridge, MA.

Gao, W. and Liu, S. (2011) 'A modified artificial bee colony algorithm', *Computers & Operations Research*, Vol. 39, No. 3, pp.687–697.

Goldberg, D.E. (1989) 'Genetic algorithms in search, optimization, and machine learning', *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

Haijun, D. and Qingxian, F. (2008) 'Bee colony algorithm for the function optimization', *Science Paper Online*, August.

Horng, M.H. and Jiang, T.W. (2011) 'Image vector quantization algorithm via honey bee mating optimization', *Expert Systems with Applications*, Vol. 38, No. 3, pp.1382–1392.

Karaboga, D. and Akay, B. (2009) 'A comparative study of artificial bee colony algorithm', *Applied Mathematics and Computation*, Vol. 214, No. 1, pp.108–132.

Karaboga, D. and Basturk, B. (2007) 'Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems', *LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, Vol. 4529, pp.789–798, Springer- Verlag, IFSA.

Karaboga, D. and Basturk, B. (2007) 'A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp.459–471.

Karaboga, D. and Basturk, B. (2008) 'On the performance of artificial bee colony algorithm', *Applied Soft Computing*, Vol. 8, No. 1, pp.687–697.

Karaboga, D., Akay, B. and Ozturk, C. (2007) 'Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks', *Modeling Decisions for Artificial Intelligence*, Vol. 4617, pp.318–329, Springer-Verlag, MDAI.

Karaboga, D. (2005) 'An idea based on honey bee swarm for numerical optimization', Techn. Rep. TR06, Erciyes Univ. Press, Erciyes.

Karaboga, N. (2009) 'A new design method based on artificial bee colony algorithm for digital iir filters', *Journal of the Franklin Institute*, Vol. 346, No. 4, pp.328–348.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Vol. 4, pp.1942–1948, IEEE.

Linh, N.T. and Anh, N.Q. (2010) 'Application artificial bee colony algorithm for reconfiguring distribution network', in *2010 Second International Conference on Computer Modeling and Simulation*, pp.102–106, IEEE.

Ma, M., Liang, J., Guo, M., Fan, Y. and Yin, Y. (2011) 'Sar image segmentation based on artificial bee colony algorithm', *Applied Soft Computing*, Vol. 11, No. 8, pp.5205–5214.

Mala, D.J., Mohan, V. and Kamalapriya, M. (2010) 'Automated software test optimisation framework-an artificial bee colony optimisation-based approach', *Software, IET*, Vol. 4, No. 5, pp.334–348.

Passino, K.M. (2002) 'Biomimicry of bacterial foraging for distributed optimization and control', *Control Systems Magazine, IEEE*, Vol. 22, No. 3, pp.52–67.

Pawar, P., Rao, R. and Shankar, R. (2008) 'Multi-objective optimization of electro-chemical machining process parameters using artificial bee colony algorithm', *Advances in Mechanical Engineering (AME-2008)*, Surat, India.

Price, K.V., Storn, R.M. and Lampinen, J.A. (2005) *Differential Evolution: A Practical Approach to Global Optimization*, Springer Verlag, Berlin Heidelberg.

Price, K.V. (1996) 'Differential evolution: a fast and simple numerical optimizer', in *Fuzzy Information Processing Society, NAFIPS, Biennial Conference of the North American*, pp.524–527, IEEE.

Rao, R.S., Narasimham, S.V.L. and Ramalingaraju, M. (2008) 'Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm', *International Journal of Electrical Power and Energy Systems Engineering*, Vol. 1, No. 2, pp.116–122.

Singh, A. (2009) 'An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem', *Applied Soft Computing*, Vol. 9, No. 2, pp.625–631.

Storn, R. and Price, K. (1995) 'Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces', *International Computer Science Institute – Publications – TR*.

Taplamacioglu, M.C. and Gozden, H. (2011) 'Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (avr) system', *Journal of the Franklin Institute*, in press.

Tsai, P.W., Pan, J.S., Liao, B.Y. and Chu, S.C. (2009) 'Enhanced artificial bee colony optimization', *International Journal of Innovative Computing, Information and Control*, Vol. 5, No. 12, pp.5081–5092.

Vesterstrom, J. and Thomsen, R. (2004) 'A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems', in *Evolutionary Computation, CEC2004, Congress on*, Vol. 2, pp.1980–1987, IEEE.

Williamson, D.F., Parker, R.A. and Kendrick, J.S. (1989) 'The box plot: a simple visual method to interpret data', *Annals of Internal Medicine*, Vol. 110, No. 11, p.916.

Xing, F., Fenglei, L. and Haijun, D. (2007) 'The parameter improvement of bee colony algorithm in tsp problem', *Science Paper Online*, November.

Zhu, G. and Kwong, S. (2010) 'Gbest-guided artificial bee colony algorithm for numerical function optimization', *Applied Mathematics and Computation*, Vol. 217, No. 7, pp.3166–3173.

Notes

1 As it is not possible to design a fully efficient population-based stochastic algorithm for all possible optimisation problems.