# Stability Analysis of Artificial Bee Colony Optimization Algorithm

Jagdish Chand Bansal*, Anshul Gopal †
South Asian University, New Delhi, India

Atulya K. Nagar‡
Liverpool Hope University, UK

January 9, 2018

**Abstract**

Theoretical analysis of swarm intelligence and evolutionary algorithms is relatively less explored area of research. Stability and convergence analysis of swarm intelligence and evolutionary algorithms can help the researchers to fine tune the parameter values. This paper presents the stability analysis of a famous Artificial Bee Colony (ABC) optimization algorithm using von Neumann stability criterion for two-level finite difference scheme. Parameter selection for the ABC algorithm is recommended based on the obtained stability conditions. The findings are also validated through numerical experiments on test problems.

**Keywords**
Artificial Bee Colony (ABC) Algorithm, Stability Analysis, Finite Difference Scheme, Parameter Selection, Stable Range

## 1 Introduction

Over past few years, algorithms taking inspiration from natural phenomena have attracted researchers. Particle Swarm Optimisation (PSO) algorithm [26], Artificial Bee Colony (ABC) optimization algorithm [21], Differential Evolution (DE) algorithm [37], Harmony Search Algorithm (HSA) [17], Gravitational Search Algorithm (GSA) [33] are few popular algorithms of this class. Study has shown that these algorithms are considered as efficient solver of complex optimization problems. Artificial Bee Colony (ABC) optimization algorithm is a prominent candidate in this field of nature inspired algorithm. It takes inspiration from the intelligent foraging behaviour and information sharing capability of honey bees [24][7]. Firstly it was introduced by Karaboga in 2005 for continuous optimization problems and later it was also modified to solve discrete optimization problems [25][27].

Recently various other variants of the ABC algorithm have been proposed which includes Chaotic ABC [4], ABC algorithm for multiobjective optimization problems [3], for constrained optimization problem [22] and various hybrid ABC algorithms [15][32][6]. The ABC algorithm is applied for solving various continuous and discrete optimization problems in the areas related to neural networks [23][28], structural engineering[13], assignment problem [31][8], image processing [14], network topology design [35] and forecasting stock markets [19].

Various researchers have worked on these meta-heuristic search algorithms analytically and experimentally. However, one of the important aspect related to such algorithms is to ensure that the error generated by them are bounded. Hence, stability analysis plays a significant role in the theoretical study of these algorithms. However, a little work has been done in the area of stability analysis of this class of algorithms. Attempts have been made to analyse the stability behaviour of few algorithms which includes Particle Swarm Optimisation (PSO) algorithm using

*jcbansal@gmail.com
†anshul.gopal@gmail.com
‡nagara@hope.ac.uk

Z-transformation [20] [30][36], Differential Evolution (DE) algorithm by using Lyapunov's stability theorem and von Neumann stability criterion [12][18], Bacterial Foraging Optimization (BFO) algorithm using Lyapunov's stability condition and eigen value method [11][9], Ant Colony Optimization (ACO) algorithm [1] and Gravitational Search algorithm (GSA) using passivity theorem and Lyapunov's stability condition [16].

In recent past, studies have been done for parameter selections of the ABC algorithm in order to get better optimization results [2][29]. Also effect of parameters like dimension, colony size, region scaling and effect of limit or abondment counter (AC) on the ABC algorithm has already been discussed in [2]. To the best of authors' knowledge, no attempt has yet been made for the stability analysis of the Artificial Bee Colony (ABC) optimization algorithm. This paper is an attempt to discuss the stability of the ABC algorithm for solving continuous optimization problems, using von Neumann stability criterion for two level finite difference scheme and to propose parameter selection based on the stability analysis. Stability of discrete version of ABC algorithm can be done in similar way because it uses the same position update equation in order to get new candidate solutions.

Rest of the paper is organised as follows: Original ABC algorithm is explained in section 2. Section 3 presents the motivation for mathematical analysis of the ABC algorithm and provide details of von Neumann stability criteria for two level finite difference scheme. In section 4 stability analysis of the ABC algorithm is carried out. Numerical experiments and results are discussed in section 5 and the study is concluded in section 6.

# 2 Artificial Bee Colony (ABC) Optimization Algorithm

Artificial Bee Colony (ABC) optimization algorithm is a population based optimization algorithm and uses an iterative method to achieve global maximum or minimum. The bee hive constitutes mainly three kinds of bees; employed bees, onlooker bees and scout bees. Exploitation of nectar source is done by employed bees and onlooker bees whereas scout bees explore the search region. In Artificial Bee Colony (ABC) optimization algorithm, the number of nectar sources equals the number of employed bees. Also, the number of employed bees and onlooker bees are equal.
The ABC algorithm has four phases; initialization, employed bee, onlooker bee and scout bee.

## 2.1 Intialization

Artificial Bee Colony (ABC) optimization algorithm starts with random selection of the food source which corresponds to the potential solution. The initial solutions are produced for employed bees by using the equation:

$$x_{i,j} = x_j^{min} + \mu(x_j^{max} - x_j^{min}), \qquad i = 1, 2, 3......N, \quad j = 1, 2, 3......D \tag{1}$$

where, $x_{i,j}$ is the $j^{th}$ dimension of the $i^{th}$ employed bee/food source; $x_j^{max}$ and $x_j^{min}$ are the upper and lower bounds of the $j^{th}$ parameter, respectively; $\mu$ is a random number in the range of $[0, 1]$. $N$ is the number of food source, i.e. swarm size and $D$ is the dimensionality of the considered optimization problem. Also, the abandonment counter (AC) of each employed bee is reset in this phase.

## 2.2 Employed Bee Phase

In this phase, for each employed bee a new candidate solution is produced. First, the solution of the employed bee is copied to new candidate solution ($v_i = x_i$). Then, one parameter of the solution is updated by using the equation:

$$v_{i,j} = \psi x_{i,j} + \phi(x_{i,j} - x_{r,j}) \ , \quad i, r \in \{1, 2, 3..., N\}, \qquad j \in \{1, 2, 3..., D\} \ \ and \ \ i \neq r \tag{2}$$

Here $j^{th}$ parameter is selected randomly for updation and the coefficient $\psi$ is taken as unity in original ABC algorithm. This is done by randomly selecting a candidate $x_r$ in the neighbourhood of $i^{th}$ candidate. $\phi$ is the random number ranging in the interval $[-1, 1]$, $N$ is the number of employed bee and $D$ is the dimensionality of the considered optimization problem. After finding new candidate solution and calculation of objective function value, fitness value of the candidate solutions and solutions of employed bees are calculated as given below:

$$fit_i = \left\{ \begin{array}{ll} \frac{1}{1+f_i}, & \text{if } f_i \geq 0 \\ 1 + abs(f_i), & \text{otherwise} \end{array} \right\}$$

where $fit_i$ is the fitness value of $i^{th}$ candidate solution, $f_i$ is the objective function value of $i^{th}$ employed bee. If the fitness value of the updated candidate solution is better than the fitness value of current solution then the current solution is replaced with candidate solution and the abandonment counter of the employed bee is reset to zero, otherwise abandonment counter is increased by one.

### 2.3 Onlooker Bee Phase

In ABC algorithm, to improve the solution each onlooker bee selects an employed bee. The probability of selecting $i^{th}$ employed bee is calculated using Roulette wheel selection:

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} \tag{3}$$

where $p_i$ is the probability of selecting $i^{th}$ employed bee. The solution of selected employed bee is improved by the onlooker bees by using equation (2). If the fitness value of new solution found by the onlooker bee is better than the employed bee, the onlooker bee is changed with the employed bee and the abandonment counter of the employed bee is reset to zero otherwise, abandonment counter is increased by one.

### 2.4 Scout Bee Phase

The abandonment counters of all employed bees are checked with a predefined limit. The employed bee, which fails to improve the solution before reaching the limit, becomes scout bee. Hereafter, equation (1) is used to the produce solution for scout bee and the abandonment counter is reset. The scout bee then becomes employed bee. Therefore, scout bee prevents the stagnation of employed bee population.

## 3 Motivation and Stability Criteria

### 3.1 Motivation

Nature inspired optimization algorithms are used to solve real world optimization problems. The algorithms provide near optimal solution and therefore the error in the obtained solution should be bounded. It is not necessary that the error generated by an iteration is always less than that of the previous iteration. It may increase indefinitely. In this case, we say that the iterative procedure is unstable. Therefore, it is worth to investigate the stability of the algorithm to bound the generated error. In the considered ABC algorithm, greedy selection is applied in both employed bee phase and onlooker bee phase which plays a vital role in reducing the error. In addition to that candidate solution is selected using fitness based probability, which also reduces the error. But upto authors' knowledge, no attempt has yet been made to restrict the error by deriving conditions on the parameters $\phi$ and $\psi$ given in equation (2). The position update equation of ABC algorithm consists of two user defined parameters $\phi$ and $\psi$. Therefore apart from greedy selection and fitness based probability selection, the value of $\phi$ and $\psi$ can also play a significant role in the ABC search process. Hence the study of finding most suitable stable range for $\phi$ and $\psi$ is important. Further with experimental results on benchmark test problems, we can analyse that though selection of candidate solution based on fitness based probability criteria and greedy selection helps in reducing error but with proper selection of parameters $\phi$ and $\psi$ the error can be reduced in less number of iterations as compared to the case when there is no restriction on these parameters. The study can serve as a recommendation to set the range of parameters $\phi$ and $\psi$ for any proposal on modification of ABC algorithm.

Section 4 presents the stability analysis of ABC algorithm by analysing the position update equation (2) of ABC algorithm. The von Neumann stability criteria for two level finite difference scheme is applied to perform the stability analysis of equation (2) and hence of ABC algorithm.

Stable range of parameter $\phi$ is defined as the range for which the ABC algorithm is stable. The next subsection explains the von Neumann stability criterion for two level finite difference scheme.

## 3.2 von Neumann Stability Criterion

The linear partial differential equation considered in this paper for an initial value problem will be represented by:

$$\frac{\partial u}{\partial t} + L_x(u) = 0$$

where $L_x(u)$ represents a linear spatial differential operator and $u$ is the dependent variable, which is a function of independent variables $x$ and $t$.
A two-level difference scheme for this linear partial differential equation can be written in the form [39]:

$$\sum_{q=-m_l}^{m_r} B_q u_{j+q}^{n+1} = \sum_{q=-n_l}^{n_r} A_q u_{j+q}^{n} \tag{4}$$

where $m_l, m_r, n_l$ and $n_r$ are non-negative integers. $j$ and $n$ represents number of grid points in the direction of $x$ and $t$ respectively.
The von Neumann stability procedure consists of replacing each term $u_j^n$ of the difference equation by $k^{th}$ Fourier component of a harmonic decomposition of $u_j^n$, i.e. by $v^n(k)e^{\iota kj\Delta x}$, where $v^n(k)$ denotes the $k^{th}$ Fourier coefficient and $\iota = \sqrt{-1}$.
The $(n)^{th}$ and $(n+1)^{th}$ Fourier coefficients of harmonic decomposition of $u_j^n$ are related by

$$v^{n+1}(k) = g(k)v^n(k)$$

where $g(k)$ is the amplification factor of the finite difference scheme.

For a two level finite difference scheme with only one dependent variable, the necessary and sufficient condition for stability is $|g(k)| \leq 1$ for all values of $k$. If $|g(k)| = 1$ for all $k$, then the difference scheme is said to be nondissipative or marginally stable, and if $|g(k)| > 1$ for some $k$, the scheme is unstable [34]. The stability criterion for finite difference scheme (4) can also be presented as below.

For two-level schemes, the square of the magnitude of amplification factor, i.e. $|g(k)|^2$ can always be expressed as a rational function given below [39]:

$$|g(k)|^2 = 1 - 4z^r \frac{S(z)}{P(z)} \tag{5}$$

where,

$$z = sin^2(\theta/2), \qquad \theta = k\Delta x \tag{6}$$

$$S(z) = \sum_{i=0}^{s} \alpha_i z^i, \qquad \alpha_0 = S(0) \neq 0 \tag{7}$$

$$P(z) = \sum_{i=0}^{d} \beta_i z^i > 0, \qquad \beta_0 = P(0) = 1 \tag{8}$$

Here, $\alpha_0, \alpha_1, ..., \alpha_s$ and $\beta_0, \beta_1, ..., \beta_d$ are constants. $r$ is a positive integer and $s$ is a non-negative integer, and they are related by the formula $r + s = m$. Where, $m = max(m_l + m_r, n_l + n_r)$ is determined by the number of spatial grid points to the left and right of $x_j$ in the difference scheme. The integer $d$ is non-negative and $d = m_r + m_l$.
The polynomial $S(z)$ determines the stability of the given difference scheme. The necessary and sufficient condition for stability is [39]:

$$S(0) > 0\ , \qquad S(z) \geq 0 \qquad for \quad 0 < z = sin^2(\theta/2) \leq 1 \tag{9}$$

In short, the necessary and sufficient condition for the stability of a difference scheme given in (4), with magnitude of amplification factor represented by $(5) - (8)$ can be stated as below:
Two level difference scheme (4) is stable iff:

$$S(0) > 0\ , \qquad S(z) \geq 0 \qquad for \quad 0 < z = sin^2(\theta/2) \leq 1 \tag{10}$$

If the solution update mechanism in an iterative algorithm is represented in the form of two level difference scheme (4), then the necessary and sufficient condition for algorithm's stability can be given by (10).

In the next section, stability analysis has been carried out for Artificial Bee Colony (ABC) optimization algorithm based on the von Neumann stability criterion discussed above.

# 4 Stability Analysis of ABC Algorithm

The parameter $\phi$ plays an important role in the search process of Artificial Bee Colony (ABC) optimization algorithm:
This parameter provides stochastic search in ABC and thus prevents the particles from getting stagnated at local optima.

In Artificial Bee Colony (ABC) optimization algorithm, the solutions are updated using the position update equation (2):

$$v_{i,j} = \psi x_{i,j} + \phi(x_{i,j} - x_{r,j}) \;, \quad i, r \in \{1, 2, 3..., N\}; \qquad j \in \{1, 2, 3..., D\} \;\; and \;\; i \neq r \tag{11}$$

If the current iteration counter is $t$, then $x_{i,j}$ is the $j^{th}$ dimension of the $i^{th}$ candidate solution at iteration $t$, while $v_{i,j}$ is the $j^{th}$ dimension of the $i^{th}$ candidate solution at iteration $(t+1)$. In the original version of ABC algorithm, the coefficient $\psi$ is taken to be unity. Without loss of generality the above equation can be written as:

$$x_{i,j}(t+1) = \psi x_{i,j}(t) \; + \; \phi(x_{i,j}(t) - x_{r,j}(t)) \tag{12}$$

In ABC algorithm, the position update equation (12) is implemented component wise, i.e. each dimension of the solution is updated independently. The only link between the dimensions of the problem space is introduced via objective function. Thus, without loss of generality, the algorithm description can be reduced for analysis purposes to the one dimension case as considered in [38][16][11]. Thus equation (12) can be written as:

$$x_i(t+1) = (\psi + \phi)x_i(t) - \phi x_r(t) \tag{13}$$

where, $r$ is randomly selected solution index different from $i$. Since $r$ and $i$ are non-negative integers in $[1, N]$, thus we can write $r = i \pm a$ where, $a$ is random positive integer in the range $[1, N]$.

Hence, equation (13) in the form of difference equation can be written as:

$$x_i^{t+1} = (\psi + \phi)x_i^t - \phi x_{i\pm a}^t \tag{14}$$

If the true solution to a problem in an $i$-$t$ computational domain is represented by $x = x(i, t)$, the approximate solution on the nodes of a computational grid will be represented by $x_{l,n} = x(i_l, t_n)$. In terms of grid points $x_{l,n}$ the difference equation can further be written as:

$$x_l^{n+1} = (\psi + \phi)x_l^n - \phi x_{l\pm a}^n \tag{15}$$

In accordance with von Neumann stability procedure, each term $x_l^n$ of the difference equation is replaced by $v^n(k)e^{\iota(kl\Delta i)}$, which is the $k^{th}$ Fourier component of a harmonic decomposition of $x_l^n$. Here, $v^n(k)$ is the $k^{th}$ Fourier coefficient in this decomposition. The $(n)^{th}$ and $(n+1)^{th}$ Fourier coefficients of harmonic decomposition of $x_l^n$ are related by:

$$v^{n+1}(k) = g(k)v^n(k) \tag{16}$$

where $g(k)$ is the amplification factor of the finite difference scheme (15).
The amplification factor of the difference scheme (15) can easily be calculated and is given by [See Appendix A]:

$$g(k) = (\psi + \phi) - \phi e^{\iota((\pm a)k\Delta i)} \tag{17}$$

or

$$g(k) = (\psi + \phi) - \phi e^{\iota(\theta)}, \qquad where \quad \theta = (\pm ak\Delta i) \tag{18}$$

Since the considered two level finite difference scheme given by equation (15) has only one dependent variable, and as discussed in section 3.2, the necessary and sufficient condition for stability is given by: $|g(k)| \leq 1$. From equation (18) we obtain [See Appendix B]

$$|g(k)| = \sqrt{\psi^2 + 4\phi(\psi+\phi)sin^2(\theta/2)} \tag{19}$$

Further using necessary and sufficient condition for stability we get

$$\sqrt{\psi^2 + 4\phi(\psi+\phi)sin^2(\theta/2)} \leq 1 \tag{20}$$

or

$$\psi^2 + 4\phi(\psi+\phi)sin^2(\theta/2) \leq 1 \tag{21}$$

or

$$\psi^2 \leq 1 - 4\phi(\psi+\phi)sin^2(\theta/2) \tag{22}$$

or

$$\psi^2 \leq 1 - 4\phi(\psi+\phi) \qquad Since, \quad sin^2(\theta/2) \leq 1 \tag{23}$$

Since, $x_{r,j}(t)$ represents the randomly selected candidate solution in the current iteration and outcome of its difference with $x_{i,j}(t)$ in equation (12) can be either positive or negative. Hence, without loss of generality equation (12) can be rewritten as:

$$x_{i,j}(t+1) = \psi x_{i,j}(t) \ + \ \phi(x_{r,j}(t) - x_{i,j}(t)) \tag{24}$$

which can again be written as

$$x_{i,j}(t+1) = \psi x_{i,j}(t) \ - \ \phi(x_{i,j}(t) - x_{r,j}(t)) \tag{25}$$

As done earlier the amplification factor is given by

$$g(k) = (\psi - \phi) + \phi e^{\iota((\pm a)k\Delta i)} \tag{26}$$

or

$$g(k) = (\psi - \phi) + \phi e^{\iota(\theta)}, \qquad where \quad \theta = (\pm ak\Delta i) \tag{27}$$

By doing the similar mathematical analysis and calculations as done earlier, the necessary and sufficient condition for the stability is given by :

$$\psi^2 \leq 1 + 4\phi(\psi-\phi) \qquad Since, \quad sin^2(\theta/2) \leq 1 \tag{28}$$

Equation (23) and (28) provide the necessary and sufficient conditions for the stability of ABC position update equation (2).

In the next subsection, a special case, i.e. ($\psi = 1$) is considered which corresponds to the original ABC algorithm. The stable range of $\phi$ is recommended for this case as well.

## 4.1 Stability analysis of ABC algorithm with coefficient $\psi = 1$

We will find the stable range of $\phi$ by two different methods. Firstly, by taking $\psi = 1$ in equation (23) and (28) to get the desired range of $\phi$. Secondly, by using the stability condition from equation (10).

By taking $\psi = 1$ in equation (23) we get
$-\phi(1+\phi) \geq 0, \qquad i.e. \ \ \phi \in [-1, 0]$

Similarly, By taking $\psi = 1$ in equation (28) we get
$\phi(1-\phi) \geq 0, \qquad i.e. \ \ \phi \in [0, 1]$

By combining the above two results we can conclude that ABC algorithm is stable for $\phi \in [-1, 1]$.

Now, we will use the stability condition as explained in equation (10) to verify the results obtained from first method.
In Artificial Bee Colony (ABC) optimization algorithm, the solutions are updated using the position update equation (2) ($\psi$ is taken as unity):

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{r,j}) \ , \quad i, r \in \{1, 2, 3..., N\}; \quad j \in \{1, 2, 3..., D\} \ \ and \ \ i \neq r \tag{29}$$

If the current iteration counter is $t$, then $x_{i,j}$ is the $j^{th}$ dimension of the $i^{th}$ candidate solution at iteration $t$, while $v_{i,j}$ is the $j^{th}$ dimension of the $i^{th}$ candidate solution at iteration $(t+1)$. Without loss of generality the above equation can be written as

$$x_{i,j}(t+1) = x_{i,j}(t) \; + \; \phi(x_{i,j}(t) - x_{r,j}(t)) \tag{30}$$

In ABC algorithm, the position update equation (12) is implemented component wise, i.e. each dimension of the solution is updated independently. The only link between the dimensions of the problem space is introduced via objective function. Thus, without loss of generality, the algorithm description can be reduced for analysis purposes to the one dimension case as considered in [38][16][11]. Thus equation (30) can be written as:

$$x_i(t+1) = (1+\phi)x_i(t) - \phi x_r(t) \tag{31}$$

where, $r$ is randomly selected solution index different from $i$. Since $r$ and $i$ are non-negative integers in $[1, N]$, thus we can write $r = i \pm a$ where, $a$ is random positive integer in the range $[1, N]$.

Hence, equation (31) in the form of difference equation can be written as:

$$x_i^{t+1} = (1+\phi)x_i^t - \phi x_{i\pm a}^t \tag{32}$$

If the true solution to a problem in an $i$-$t$ computational domain is represented by $x = x(i, t)$, the approximate solution on the nodes of a computational grid will be represented by $x_{l,n} = x(i_l, t_n)$. In terms of grid points $x_{l,n}$ the difference equation can further be written as:

$$x_l^{n+1} = (1+\phi)x_l^n - \phi x_{l\pm a}^n \tag{33}$$

In accordance with von Neumann stability procedure, each term $x_l^n$ of the difference equation is replaced by $v^n(k)e^{\iota(kl\Delta i)}$, which is the $k^{th}$ Fourier component of a harmonic decomposition of $x_l^n$. Here, $v^n(k)$ is the $k^{th}$ Fourier coefficient in this decomposition. The $(n)^{th}$ and $(n+1)^{th}$ Fourier coefficients of harmonic decomposition of $x_l^n$ are related by:

$$v^{n+1}(k) = g(k)v^n(k) \tag{34}$$

where $g(k)$ is the amplification factor of the finite difference scheme (33).

The amplification factor of the difference scheme (33) can easily be calculated and is given by [See Appendix A]:

$$g(k) = (\psi + \phi) - \phi e^{\iota(\theta)}, \qquad where \qquad \theta = (\pm ak\Delta i) \tag{35}$$

Further, we can obtain

$$|g(k)|^2 = (1+\phi)^2 + (\phi)^2 - 2\phi(w - \phi)cos\theta \tag{36}$$

$$|g(k)|^2 - 1 = 4\phi(1+\phi)sin^2(\theta/2) \tag{37}$$

where, $\theta = (\pm ak\Delta i)$

For two-level schemes, the modulus of square of amplification factor can be expressed in the form of rational function as:

$$|g(k)|^2 = 1 - 4z^r \frac{S(z)}{P(z)} \tag{38}$$

where,
$z = sin^2(\theta/2)$ , $\theta = (k\Delta x)$
$S(z) = \sum_{i=0}^{s} \alpha_i z^i$ , $\alpha_0 = S(0) \neq 0$
$P(z) = \sum_{i=0}^{d} \beta_i z^i$ , $\beta_0 = P(0) = 1$
Here, $r$ is a positive integer, $s$ and $d$ are non negative integers.

The necessary and sufficient condition for stability as given in equation (10) is [39]:

1. $S(0) > 0$
2. $S(z) \geq 0$ for $0 < z = sin^2(\theta/2) \leq 1$

By comparing equation (37) with equation (38) we get:
$S(z) = -\phi(1 + \phi)$

Hence, the necessary and sufficient condition for stability is given by:

$$-\phi(1 + \phi) \geq 0, \qquad i.e. \;\; \phi \in [-1, 0]$$

Since, $x_{r,j}(t)$ represents the randomly selected candidate solution in the current iteration and outcome of its difference with $x_{i,j}(t)$ in equation (12) can be either positive or negative. Hence, without loss of generality equation (12) can be rewritten as:

$$x_{i,j}(t + 1) = x_{i,j}(t) \; + \; \phi(x_{r,j}(t) - x_{i,j}(t)) \tag{39}$$

which can again be written as:

$$x_{i,j}(t + 1) = x_{i,j}(t) \; - \; \phi(x_{i,j}(t) - x_{r,j}(t)) \tag{40}$$

By doing the similar mathematical analysis and calculations as done earlier, the necessary and sufficient condition for the stability of ABC algorithm is given by :

$$S(z) = \phi(1 - \phi) \geq 0 \qquad i.e. \;\; \phi \in [0, 1]$$

So, by combining the above two results, we can easily interpret that necessary and sufficient condition for the stability of update equation of ABC algorithm is that the parameter $\phi$ must lie in the interval $[-1, 1]$. In short, the stability of ABC algorithm depends upon the range of parameter $\phi$.

The findings of the stability analysis of Artificial Bee Colony (ABC) optimization algorithm verifies the recommended setting of parameter $\phi$ in Artificial Bee Colony (ABC) algorithm search process. That is, for the stability of ABC algorithm, $\phi$ must be in the range $[-1, 1]$.

This stability analysis can further be applied to improve the performance of various advanced variants of ABC, e.g. [5][10]. In [5], Akay et al. introduced an adaptive scaling factor '$\phi$' based on *Rechenbergs 1/5th mutation rule*. The analysis undertaken in this paper can be used to set the value of coefficient '$\psi$'(refer equation 2) as a function of adaptive scaling factor (ASF). This strategy can further improve the ABC algorithm in terms of accuracy.

Similarly, the stability analysis carried out in this paper can also be applied to set the scaling factors '$\phi_G$' and '$\phi_C$' introduced by Das et al. [10] in two different ways:

1. Stability analysis of *ABC* with two scaling factors '$\phi_G$' and '$\phi_C$' can provide a relation between '$\phi_G$' and '$\phi_C$'.
2. A relation like equation (23) and (28) among '$\psi$', '$\phi_G$' and '$\phi_C$' can be obtained.

An intensive future research is required for these analyses. Other variants of ABC can also be checked for this kind of relations between proposed coefficient $\psi$ and scaling factor $\phi$.

Next section verifies numerically that other ranges of $\phi$ are not as efficient as stable range $[-1, 1]$.

# 5 Numerical Experiments

In order to validate our theoretical findings numerically, four different ranges of parameter $\phi$ are considered. Numerical experiments are performed with $\phi \in [-1, 1]$, $\phi \in [-3, -1]$, $\phi \in [1, 3]$ and $\phi \in [-2, -1] \cup [1, 2]$. The length of all the ranges are same as the length of stable range. The range $[-3, -1]$ represents the case if $\phi$ is selected from left of stable range, while the range $[1, 3]$ chooses $\phi$ from the right of stable range. The range $[-2, -1] \cup [1, 2]$ represents choice of selection of $\phi$ from left or right of the stable range.

Following three types of numerical experiments are performed:

1. In subsection 5.1, error convergence graphs are plotted .
2. Subsection 5.2, presents evolving performance of the ABC algorithm for four different ranges of parameter $\phi$ and results are validated through Mann-Whiteny U rank sum test.
3. Effect of $\phi$ range variation on efficiency of the ABC algorithm is discussed in section 5.3.

**Table 1**
List of Test Problems (**AE**: Acceptable Error, **U**: Uni-modal, **M**: Multi-modal, **S**: Separable, **N**: Non-separable )

| Name of the problem | Objective function | Search Range | Optimum Value | Dim (n) | AE | Characteristic |
|---|---|---|---|---|---|---|
| Sphere | $Minf_1(x) = \sum_{i=1}^{n} x_i^2$ | $[-5.12, 5.12]$ | $f(\vec{0}) = 0$ | 30 | $8.0E-04$ | U, S |
| Rastrigin | $Minf_2(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$ | $[-5.12, 5.12]$ | $f(\vec{0}) = 0$ | 30 | $5.0E-01$ | M, S |
| Griewank | $Minf_3(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$ | $[-600, 600]$ | $f(\vec{0}) = 0$ | 30 | $9.05E-06$ | M, N |
| Alpine | $Minf_4(x) = \sum_{i=1}^{n} \lvert x_i \sin x_i + (0.1)x_i \rvert$ | $[-10, 10]$ | $f(\vec{0}) = 0$ | 10 | $8.5E-04$ | M, S |
| Cosine Mixture | $Minf_5(x) = \sum_{i=1}^{n} {x_i}^2 - 0.1(\sum_{i=1}^{n}\cos 5\pi x_i) + 0.1n$ | $[-1, 1]$ | $f(\vec{0}) = -n \times 0.1$ | 30 | $6.3E-06$ | M, S |
| Zakharov | $Minf_6(x) = \sum_{i=1}^{n} {x_i}^2 + (\sum_{i=1}^{n} \frac{ix_i}{2})^2 + (\sum_{i=1}^{n} \frac{ix_1}{2})^4$ | $[-5.12, 5.12]$ | $f(\vec{0}) = 0$ | 30 | 135.0 | M, N |
| Axis parallel hyper-ellipsoid | $Minf_7(x) = \sum_{i=1}^{n} i.x_i^2$ | $[-5.12, 5.12]$ | $f(\vec{0}) = 0$ | 30 | $6.5E-06$ | U, S |
| Sum of different powers | $Minf_8(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$ | $[-1, 1]$ | $f(\vec{0}) = 0$ | 30 | $1.0E-05$ | U, S |
| Rosenbrock | $Minf_9(x) = \sum_{i=1}^{n}(100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$ | $[-30, 30]$ | $f(\vec{0}) = 0$ | 30 | 25.0 | U, N |
| Shifted Ackley | $Minf_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)) + 20 + e + f_{bias}$, $z = (x - o)$, $x = (x_1, x_2, ........x_n)$, $o = (o_1, o_2, ........o_n)$ | $[-32, 32]$ | $f(o) = f_{bias} = -140$ | 10 | 20.0 | M, N |

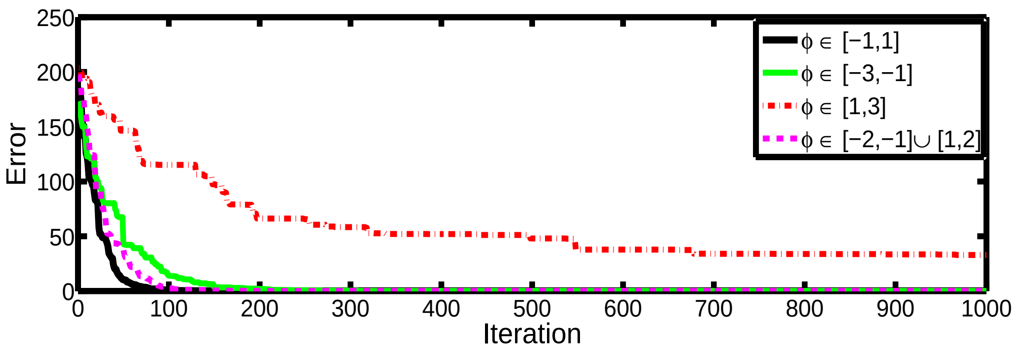


**Fig. 1.** Convergence graphs for $f_1$ with various ranges of $\phi$.

## 5.1 Convergence

In order to see convergence behaviour of the ABC algorithm with different ranges of parameter $\phi$, experiments are performed over 10 test problems given in Table 1. The test problems of Table 1 consists of separable, non separable, uni-modal and multi-modal optimization functions.

Figure 1 to 10 show convergence graphs of the ABC algorithm over 1000 iterations. It can be concluded that in ABC algorithm greedy selection and fitness based probability selection is done to get improved results in each iteration hence, error is decreasing with increase in iteration for almost every range of parameter $\phi$. Graphs clearly show that for theoretically proposed stable range of parameter $\phi$, i.e. [-1,1], the algorithm takes less number of iterations to reach to near optimal solution as compared to other ranges of $\phi$. Hence the proposed stability criteria for ABC algorithm plays a vital role in convergence behaviour of ABC algorithm.

## 5.2 Evolving Performance of ABC Algorithm

Accuracy of the ABC algorithm is tested for various ranges of parameter $\phi$ using evolving performance indicator based on mean error. Following parameter settings are adopted to perform the numerical experiments on the test problems given in Table 1.

1. Swarm size: 50
2. Maximum number of runs: 100
3. Maximum number of iterations: 1000
4. Acceptable error: Refer Table 1

Numerical results are presented in Table 2. Table 2 presents the mean error in 200, 400, 600, 800 and 1000 iterations over 100 runs. This mean error is obtained for all four considered ranges

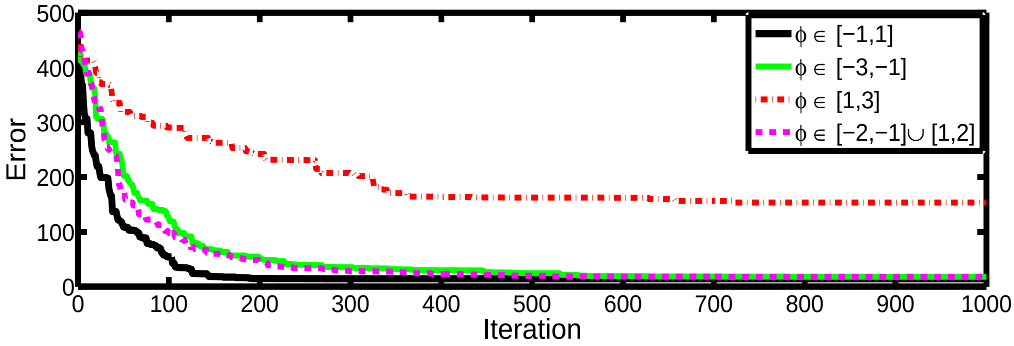


**Fig. 2.** Convergence graphs for $f_2$ with various ranges of $\phi$.

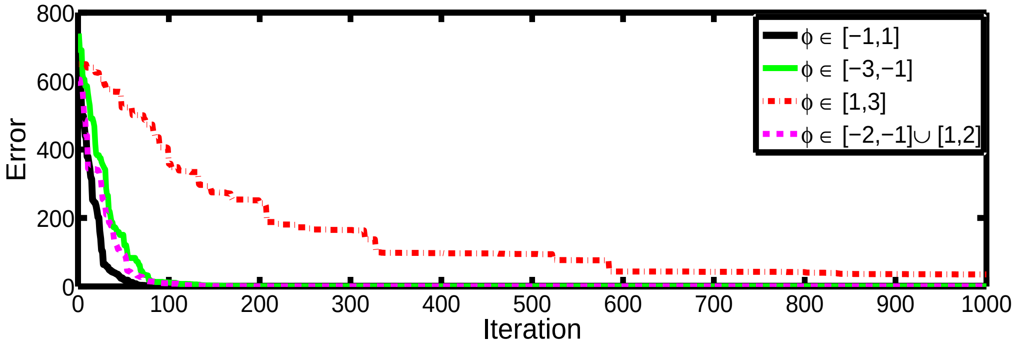


**Fig. 3.** Convergence graphs for $f_3$ with various ranges of $\phi$.

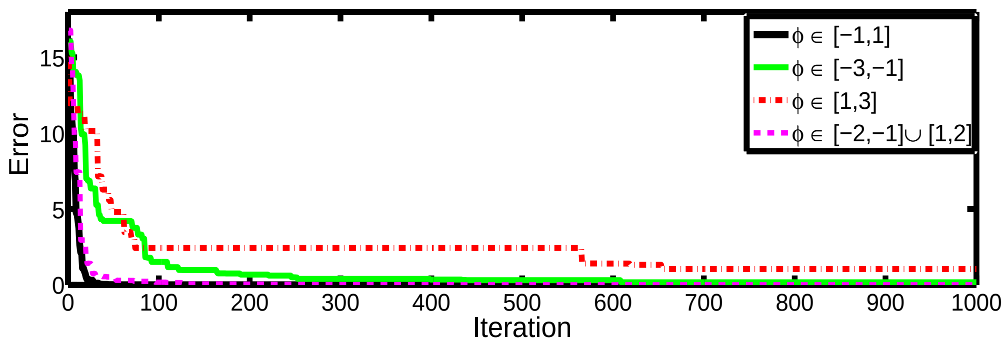


**Fig. 4.** Convergence graphs for $f_4$ with various ranges of $\phi$.

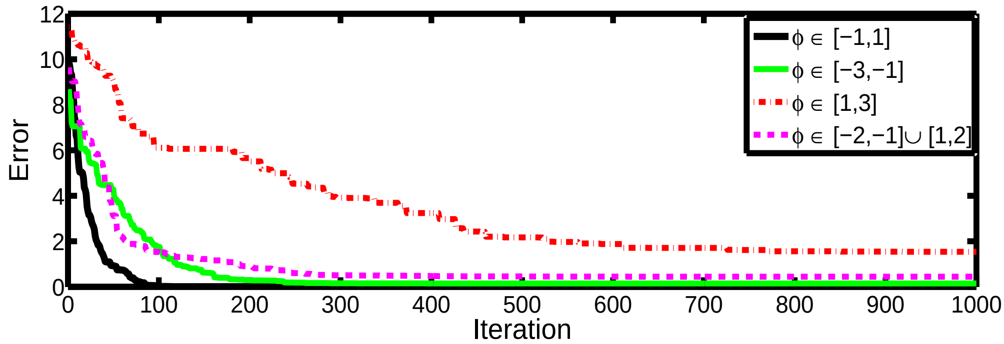


**Fig. 5.** Convergence graphs for $f_5$ with various ranges of $\phi$.

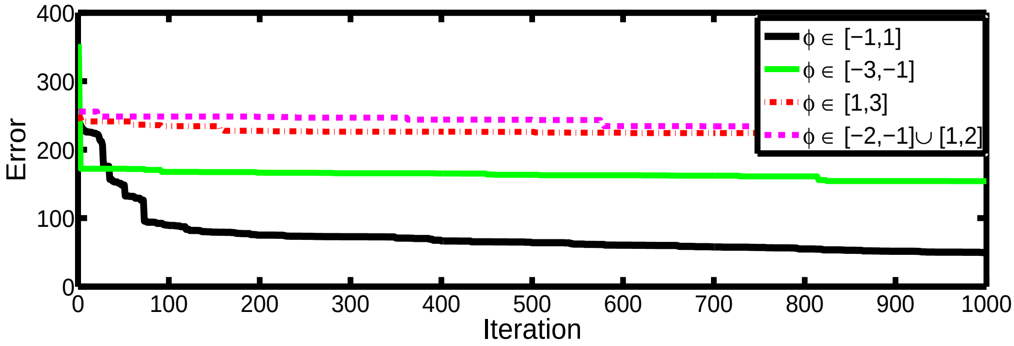


**Fig. 6.** Convergence graphs for $f_6$ with various ranges of $\phi$.

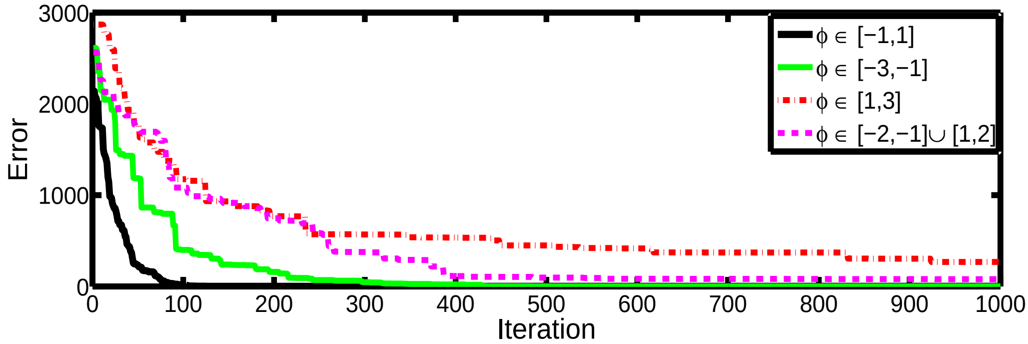


**Fig. 7.** Convergence graphs for $f_7$ with various ranges of $\phi$.

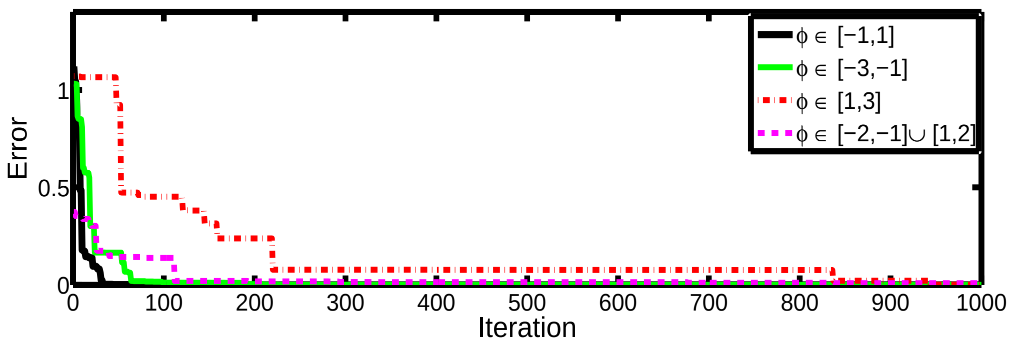


**Fig. 8.** Convergence graphs for $f_8$ with various ranges of $\phi$.

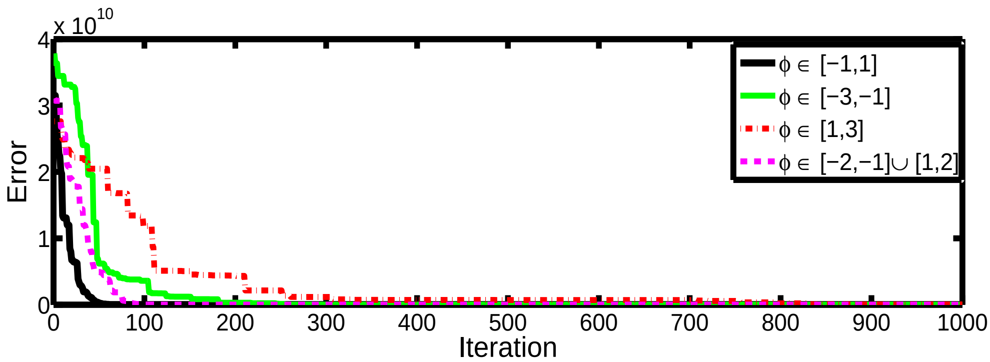


**Fig. 9.** Convergence graphs for $f_9$ with various ranges of $\phi$.

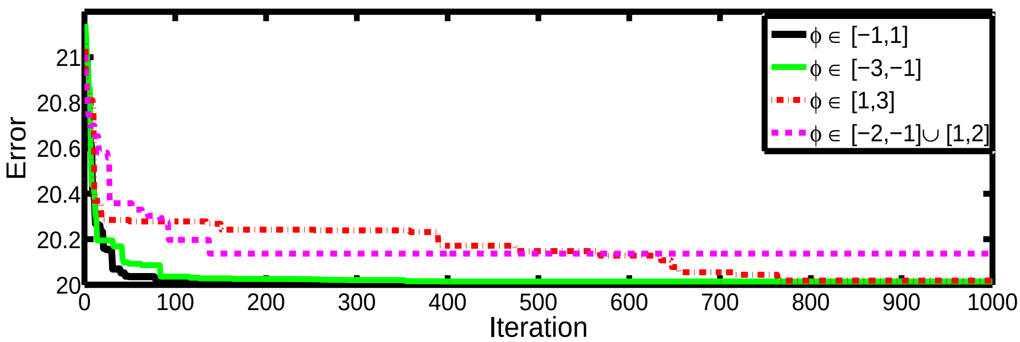


**Fig. 10.** Convergence graphs for $f_{10}$ with various ranges of $\phi$.

of parameter $\phi$. From Table 2, it can be observed that the minimum mean error is obtained if $\phi \in [-1, 1]$.

In order to check whether this difference in mean error is due to randomness or not, a non-parametric test, Mann-Whiteny U rank sum test is applied. In this study, the test is performed on mean error at 5% level of significance between *ABC* with stable range and ABC with $\phi \in [-3, -1]$, $\phi \in [-2, -1] \cup [1, 2]$ and $\phi \in [1, 3]$. Table 4 represents results of Mann-Whiteny U rank sum test for mean error (ME) over 100 runs.

If these data sets have not significant difference then we say that null hypothesis (there is no significant difference) is accepted and '=' sign appears. If the difference is significant then we say that null hypothesis is rejected. In Table 4, '+' sign appears if the variant of ABC algorithm with stable range performs better than other variants. Otherwise, '-' sign will appear. Table 4 contains 9 '+' signs out of 10. Thus it is clear that the accuracy of ABC algorithm is better if the parameter $\phi$ is in the stable range.

**Table 2**
Mean Error (**ME**) for various ranges of $\phi$ (**TP**: Test Problem)

| **TP** | **Number of Iterations** | **ME for** $\phi \in [-1, 1]$ | **ME for** $\phi \in [-3, -1]$ | **ME for** $\phi \in [1, 3]$ | **ME for** $\phi \in [-2, -1] \cup [1, 2]$ |
|---|---|---|---|---|---|
| | 200 | **0.0198** | 0.0829 | 61.5894 | 29.1486 |
| | 400 | **2.03E-05** | 0.00017 | 33.0215 | 15.2819 |
| $f_1$ | 600 | **4.47E-08** | 2.98E-07 | 19.6312 | 9.2063 |
| | 800 | **6.4E-11** | 4.14E-10 | 14.1740 | 6.2110 |
| | 1000 | **5.44E-14** | 2.97E-13 | 9.8760 | 4.5989 |
| | 200 | **33.3116** | 45.5243 | 215.445 | 123.4919 |
| | 400 | **8.9279** | 13.7808 | 163.3054 | 95.2967 |
| $f_2$ | 600 | **3.2563** | 6.4717 | 139.3662 | 84.2060 |
| | 800 | **1.2583** | 3.2553 | 124.4344 | 79.9609 |
| | 1000 | **0.20008** | 1.5042 | 115.8064 | 77.2762 |
| | 200 | **0.1056** | 0.5024 | 224.1136 | 114.6292 |
| | 400 | **0.0008** | 0.0089 | 114.6387 | 80.2577 |
| $f_3$ | 600 | **2.66E-05** | 0.00029 | 67.5725 | 52.8610 |
| | 800 | **7.0E-07** | 8.79E-06 | 46.1771 | 35.7186 |
| | 1000 | **1.29E-08** | 1.86E-07 | 33.7703 | 26.2357 |
| | 200 | **1.2417** | 2.2330 | 23.9806 | 10.8493 |
| | 400 | **0.1056** | 0.1876 | 16.8943 | 7.7951 |
| $f_4$ | 600 | **0.0173** | 0.0254 | 14.0703 | 6.8470 |
| | 800 | **0.00441** | 0.00450 | 12.5115 | 6.3361 |
| | 1000 | **0.00065** | 0.0011 | 11.5609 | 6.1795 |
| | 200 | **0.1566** | 0.2616 | 4.5688 | 2.3293 |
| | 400 | **0.00020** | 0.00063 | 2.8025 | 1.4782 |
| $f_5$ | 600 | **2.2E-07** | 8.76E-07 | 2.0048 | 1.1653 |
| | 800 | **1.43E-10** | 7.28E-10 | 1.5822 | 1.0061 |
| | 1000 | **9.9E-14** | 5.82E-13 | 1.3401 | 0.9067 |

**Table 2 Continued:**

| TP | Number of Iterations | ME for $\phi \in [-1, 1]$ | ME for $\phi \in [-3, -1]$ | ME for $\phi \in [1, 3]$ | ME for $\phi \in [-2, -1] \cup [1, 2]$ |
|---|---|---|---|---|---|
| | 200 | **161.0309** | 192.2592 | 214.5265 | 194.6106 |
| | 400 | **149.8301** | 187.1040 | 210.8385 | 188.6762 |
| $f_6$ | 600 | **141.7908** | 183.7348 | 207.3502 | 185.2423 |
| | 800 | **136.2172** | 179.9766 | 204.7566 | 182.8562 |
| | 1000 | **131.304** | 177.4148 | 202.4118 | 180.9851 |
| | 200 | **0.0667** | 0.6254 | 863.9613 | 433.1880 |
| | 400 | **7.98E-05** | 0.0014 | 446.9480 | 199.3566 |
| $f_7$ | 600 | **3.12E-07** | 4.16E-06 | 269.4713 | 117.2702 |
| | 800 | **5.43E-10** | 7.61E-09 | 167.4189 | 84.8911 |
| | 1000 | **5.66E-13** | 8.48E-12 | 111.8956 | 62.5686 |
| | 200 | **0.00040** | 0.00099 | 0.4317 | 0.2166 |
| | 400 | **3.21E-06** | 1.67E-05 | 0.2155 | 0.1198 |
| $f_8$ | 600 | **6.57E-08** | 4.02E-07 | 0.1202 | 0.0484 |
| | 800 | **8.8E-10** | 1.19E-08 | 0.0690 | 0.0329 |
| | 1000 | **1.9E-11** | 5.51E-10 | 0.0429 | 0.0238 |
| | 200 | **3199.1690** | 36530.66 | 9.51E+09 | 4.15E+09 |
| | 400 | **361.4558** | 761.3498 | 4.1E+09 | 1.76E+09 |
| $f_9$ | 600 | **77.4787** | 235.8983 | 2.27E+09 | 1.12E+09 |
| | 800 | **21.1273** | 148.5883 | 1.3E+09 | 6.19E+08 |
| | 1000 | **7.8110** | 118.4342 | 8.14E+08 | 3.98E+08 |
| | 200 | **20.00409** | 20.01365 | 20.1471 | 20.10007 |
| | 400 | **20.00079** | 20.00404 | 20.1046 | 20.0769 |
| $f_{10}$ | 600 | **20.00027** | 20.002 | 20.0779 | 20.0606 |
| | 800 | **20.00013** | 20.00125 | 20.0631 | 20.0480 |
| | 1000 | **20.00008** | 20.0009 | 20.0483 | 20.0399 |

**Table 3**
Comparision of ABC algorithm for various ranges of $\phi$ using Mann-Whiteny U rank sum test (**TP**: Test Problem, **ME**: Mean Error, $A_1$: $\phi \in [-1, 1]$, $A_2$: $\phi \in [-3, -1]$, $A_3$: $\phi \in [1, 3]$, $A_4$: $\phi \in [-2, -1] \cup [1, 2]$ )

| **TP** | **Based on ME of** $A_1$ **Vs** $A_2$ | **Based on ME of** $A_1$ **Vs** $A_3$ | **Based on ME of** $A_1$ **Vs** $A_4$ |
|---|---|---|---|
| $f_1$ | + | + | + |
| $f_2$ | + | + | + |
| $f_3$ | + | + | + |
| $f_4$ | + | + | + |
| $f_5$ | + | + | + |
| $f_6$ | + | + | + |
| $f_7$ | + | + | + |
| $f_8$ | + | + | + |
| $f_9$ | + | + | + |
| $f_{10}$ | = | = | = |
| **Number of + sign** | **9** | **9** | **9** |

## 5.3 Efficiency

To see the effect of parameter $\phi$ over the efficiency of ABC algorithm, numerical experiments are performed.

The average number of function evaluations (AFEs) are reported in Table 3 for all four considered ranges of parameter $\phi$. Test problems of Table 1 are considered for experiments and parameter selection is same as explained in section 4.2. In addition to that, the algorithm is stopped when either acceptable error is obtained or maximum number of iterations (which is set to be 1000) has been reached. It is clear that for stable range of parameter $\phi$, ABC algorithm is most efficient.

**Table 4**
**AFEs** for various ranges of $\phi$ (**AFEs**: Average number of Function Evaluations, **TP**: Test Problem)

| TP | Range of $\phi$ | AFEs |
|---|---|---|
| $f_1$ | $[-1, 1]$ | **13181.5** |
| | $[-3, -1]$ | 17296.5 |
| | $[1, 3]$ | 50025 |
| | $[-2, -1] \cup [1, 2]$ | 30969.5 |
| $f_2$ | $[-1, 1]$ | **42517.5** |
| | $[-3, -1]$ | 49263 |
| | $[1, 3]$ | 50025 |
| | $[-2, -1] \cup [1, 2]$ | 50025 |
| $f_3$ | $[-1, 1]$ | **32399** |
| | $[-3, -1]$ | 39702.5 |
| | $[1, 3]$ | 50025 |
| | $[-2, -1] \cup [1, 2]$ | 42525 |
| $f_4$ | $[-1, 1]$ | **43441** |
| | $[-3, -1]$ | 46956 |
| | $[1, 3]$ | 50025 |
| | $[-2, -1] \cup [1, 2]$ | 48577.02 |
| $f_5$ | $[-1, 1]$ | **24624.5** |
| | $[-3, -1]$ | 26650 |
| | $[1, 3]$ | 50025 |
| | $[-2, -1] \cup [1, 2]$ | 44153.5 |
| $f_6$ | $[-1, 1]$ | **36149.5** |
| | $[-3, -1]$ | 48858.5 |
| | $[1, 3]$ | 49386.94 |
| | $[-2, -1] \cup [1, 2]$ | 48787.82 |

**Table 4 Continued:**

| **TP** | **Range of $\phi$** | **AFEs** |
|---|---|---|
| $f_7$ | $[-1,1]$ | **23449** |
| | $[-3,-1]$ | 28965.5 |
| | $[1,3]$ | 50025 |
| | $[-2,-1]\cup[1,2]$ | 38044.5 |
| $f_8$ | $[-1,1]$ | **16121.5** |
| | $[-3,-1]$ | 19639 |
| | $[1,3]$ | 50025 |
| | $[-2,-1]\cup[1,2]$ | 32579.52 |
| $f_9$ | $[-1,1]$ | **37509** |
| | $[-3,-1]$ | 49798 |
| | $[1,3]$ | 50025 |
| | $[-2,-1]\cup[1,2]$ | 49317.02 |
| $f_{10}$ | $[-1,1]$ | **6411.5** |
| | $[-3,-1]$ | 11412.66 |
| | $[1,3]$ | 50025 |
| | $[-2,-1]\cup[1,2]$ | 50025 |

As an overall observation, the performance of ABC algorithm significantly deteriorates if the range of parameter $\phi$ is deviated from the stable range $[-1,1]$. From above numerical experiments, we can say that for better convergence, accuracy and efficiency the recommended range of parameter $\phi$ is same as the stable range $[-1,1]$ obtained in section 4.

# 6 Conclusion and Future Work

Mathematical validity of parameters of probabilistic algorithms has always been a challenging task. Stability theory can help to derive value or the range of one or more parameters for the algorithms. In this paper, stability analysis of the ABC position update equation with parameter $\phi$ and coefficient $\psi$ has been carried out. A generic method for testing the stability, von Neumann stability procedure for two-level finite difference scheme is considered. The outcome of stability analysis verifies the usual settings of parameter $\phi$ in the range [-1, 1]. Also stability condition depending on parameter $\phi$ and coefficient $\psi$ is proposed which will bound the error in subsequent iterations. The findings are verified with graphical interpretation and numerical results over test problems. The study can also further be extended for the convergence analysis of ABC algorithm.


# 7 Acknowledgement

The second author acknowledges the funding from South Asian University New Delhi, India to carry out this research.


**Appendix A**

**Finding amplification factor discussed in equation (18).**

As discussed in section 3.3, amplification factor is calculated by replacing each term $x_l^n$ of the difference equation (15) by $v^n(k)e^{\iota(kl\Delta i)}$ and then using equation (16). After doing the required substitutions, equation (15) is modified as

$$v^{n+1}(k)e^{\iota(kl\Delta i)} = (\psi+\phi)v^n(k)e^{\iota(kl\Delta i)} - (\phi)v^n(k)e^{\iota(k(l\pm a)\Delta i)} \tag{41}$$

or

$$v^{n+1}(k) = (\psi+\phi)v^n(k) - (\phi)v^n(k)e^{\iota((\pm a)k\Delta i)} \tag{42}$$

or

$$v^{n+1}(k) = \left[(\psi+\phi) - (\phi)e^{\iota((\pm a)k\Delta i)}\right]v^n(k) \tag{43}$$

By comparing equation (16) and (43), the amplification factor is given by

$$g(k) = (\psi + \phi) - \phi e^{\iota(\theta)}, \qquad where \qquad \theta = ((\pm a)k\Delta i) \tag{44}$$

Similarly, amplification factor described in equation (27) can be calculated.

**Appendix B**

**Finding modulus of amplification factor discussed in equation (19).**

By using equation (18), amplification factor is given by

$$g(k) = (\psi + \phi) - \phi e^{\iota(\theta)} \tag{45}$$

or

$$g(k) = (\psi + \phi) - \phi(cos(\theta)) - \iota\phi sin(\theta) \tag{46}$$

By taking modulus of the above equation we get

$$|g(k)| = \sqrt{\big[(\psi + \phi) - \phi cos(\theta)\big]^2 + \big[\phi sin(\theta)\big]^2} \tag{47}$$

By further expansion, the above equation (47) is modified as

$$|g(k)| = \sqrt{(\psi + \phi)^2 + \phi cos(\theta)^2 - 2\phi(\psi + \phi)cos(\theta) + (\phi sin(\theta))^2} \tag{48}$$

or

$$|g(k)| = \sqrt{(\psi + \phi)^2 + \phi^2 - 2\phi(\psi + \phi)\big(1 - 2sin^2(\theta/2)\big)} \tag{49}$$

or

$$|g(k)| = \sqrt{(\psi + \phi)^2 + \phi^2 - 2\phi(\psi + \phi) + 4\phi(\psi + \phi)sin^2(\theta/2)} \tag{50}$$

or

$$|g(k)| = \sqrt{(\psi + \phi - \phi)^2 + 4\phi(\psi + \phi)sin^2(\theta/2)} \tag{51}$$

or

$$|g(k)| = \sqrt{\psi^2 + 4\phi(\psi + \phi)sin^2(\theta/2)} \tag{52}$$

Similarly, modulus of amplification factor given in equation (27) can be calculated.

# References

# References


[1] Ajith Abraham, Amit Konar, Nayan R Samal, and Swagatam Das. Stability analysis of the ant system dynamics with non-uniform pheromone deposition rules. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1103–1108. IEEE, 2007.

[2] Bahriye Akay and Dervis Karaboga. Parameter tuning for the artificial bee colony algorithm. *ICCCI*, 2009:608–619, 2009.

[3] Reza Akbari, Ramin Hedayatzadeh, Koorush Ziarati, and Bahareh Hassanizadeh. A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation*, 2:39–52, 2012.

[4] Bilal Alatas. Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37(8):5682–5687, 2010.

[5] DervisKaraboga BahriyeAkay. A modified abc algorithm for real parameter optimization. *information sciences*, 192, 2012.

[6] Anan Banharnsakun, Tiranee Achalakul, and Booncharoen Sirinaovakul. Abc-gsx: A hybrid method for solving the traveling salesman problem. In *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pages 7–12. IEEE, 2010.

[7] Jagdish Chand Bansal, Harish Sharma, and Shimpi Singh Jadon. Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, 5(1-2):123–159, 2013.

[8] Adil Baykasoğlu, Lale Özbakır, and Pınar Tapkan. Artificial bee colony algorithm and its application to generalized assignment problem. In *Swarm intelligence, focus on ant and particle swarm optimization*. InTech, 2007.

[9] Arijit Biswas, Swagatam Das, Ajith Abraham, and Sambarta Dasgupta. Stability analysis of the reproduction operator in bacterial foraging optimization. *Theoretical Computer Science*, 411(21):2127–2139, 2010.

[10] Swagatam Das, Subhodip Biswas, and Souvik Kundu. Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization. *Applied Soft Computing*, 13(12):4676–4694, 2013.

[11] Swagatam Das, Sambarta Dasgupta, Arijit Biswas, Ajith Abraham, and Amit Konar. On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(3):670–679, 2009.

[12] Sambarta Dasgupta, Swagatam Das, Arijit Biswas, and Ajith Abraham. On stability and convergence of the population-dynamics in differential evolution. *Ai Communications*, 22(1):1–20, 2009.

[13] ZH Ding, M Huang, and ZR Lu. Structural damage detection using artificial bee colony algorithm with hybrid search strategy. *Swarm and Evolutionary Computation*, 28:1–13, 2016.

[14] Amer Draa and Amira Bouaziz. An artificial bee colony algorithm for image contrast enhancement. *Swarm and Evolutionary computation*, 16:69–84, 2014.

[15] Hai-bin Duan, Chun-fang Xu, and Zhi-Hui Xing. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *International Journal of Neural Systems*, 20(01):39–50, 2010.

[16] Faezeh Farivar and Mahdi Aliyari Shoorehdeli. Stability analysis of particle dynamics in gravitational search optimization algorithm. *Information Sciences*, 337:25–43, 2016.

[17] Zong Woo Geem, Joong Hoon Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68, 2001.

[18] Anshul Gopal and Jagdish Chand Bansal. Stability analysis of differential evolution. In *Computational Intelligence (IWCI), International Workshop on*, pages 221–223. IEEE, 2016.

[19] Tsung-Jung Hsieh, Hsiao-Fen Hsiao, and Wei-Chang Yeh. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied soft computing*, 11(2):2510–2525, 2011.

[20] Visakan Kadirkamanathan, Kirusnapillai Selvarajah, and Peter J Fleming. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, 10(3):245–255, 2006.

[21] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[22] Dervis Karaboga and Bahriye Akay. A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied soft computing*, 11(3):3021–3031, 2011.

[23] Dervis Karaboga and Bahriye Basturk. Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In *International Fuzzy Systems Association World Congress*, pages 789–798. Springer, 2007.

[24] Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, and Nurhan Karaboga. A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review*, 42(1):21–57, 2014.

[25] Mina Husseinzadeh Kashan, Nasim Nahavandi, and Ali Husseinzadeh Kashan. Disabc: a new artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, 12(1):342–352, 2012.

[26] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[27] Mustafa Servet Kiran. The continuous artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, 33:15–23, 2015.

[28] Pravin Yallappa Kumbhar and Shoba Krishnan. Use of artificial bee colony (abc) algorithm in artificial neural network synthesis. *Int J Adv Eng Sci Technol*, 11(1):162–171, 2011.

[29] Guoqiang Li, Peifeng Niu, and Xingjun Xiao. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. *Applied soft computing*, 12(1):320–332, 2012.

[30] Ning Li, De-Bao Sun, Tong Zou, Yuan-Qing Qin, and Yu Wei. Analysis for a particle's trajectory of pso based on difference equation. *Jisuanji Xuebao/Chinese Journal of Computers*, 29(11):2052–2061, 2006.

[31] Magdalena Metlicka and Donald Davendra. Chaos driven discrete artificial bee algorithm for location and assignment optimisation problems. *Swarm and Evolutionary Computation*, 25:15–28, 2015.

[32] Celal Ozturk and Dervis Karaboga. Hybrid artificial bee colony algorithm for neural network training. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 84–88. IEEE, 2011.

[33] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[34] Robert D Richtmyer and KW Morton. *Different methods for initial value problems.* Interscience, 1967.

[35] Amani Saad, Salman A. Khan, and Amjad Mahmood. A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design. *Swarm and Evolutionary Computation*, 2017.

[36] Nayan R Samal, Amit Konar, Swagatam Das, and Ajith Abraham. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1769–1776. IEEE, 2007.

[37] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[38] Ioan Cristian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 85(6):317–325, 2003.

[39] RF Warming and BJ Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of computational physics*, 14(2):159–179, 1974.