Analysing convergence, consistency and trajectory of Artificial Bee Colony Algorithm

Jagdish Chand Bansal, *Member, IEEE,* Anshul Gopal and Atulya K. Nagar

Abstract

Recently, swarm intelligence based algorithms gained attention of the researchers due to their wide applicability and ease of implementation. However, much research has been made on the development of swarm intelligence algorithms but theoretical analysis of these algorithms is still a less explored area of the research. Theoretical analyses of trajectory and convergence of potential solutions towards the equilibrium point in the search space can help the researchers to understand the iteration-wise behaviour of the algorithms which can further help in making them efficient. Artificial Bee Colony (ABC) optimization algorithm is swarm intelligence based algorithm. This paper presents the convergence analysis of ABC algorithm using theory of dynamical system. Convergent boundaries for the parameters of ABC update equation have also been proposed. Also the trajectory of potential solutions in the search space is analysed by obtaining a partial differential equation corresponding to the position update equation of ABC algorithm. The analysis reveals that the ABC algorithm performs better when parameters of the update equation are in the convergent region and potential solutions movement follows 1-Dimensional advection equation.

Index Terms

Artificial Bee Colony (ABC) Algorithm, advection equation, convergence analysis, finite difference scheme, swarm intelligence.

I. Introduction

In the recent past, researchers have shown interest in algorithms inspired from natural phenomena. To name a few we have Particle Swarm Optimisation (PSO) algorithm [23] taking inspiration from birds flocking, Artificial Bee Colony (ABC) optimization algorithm [21] inspired by foraging behaviour of honey bees, Gravitational Search Algorithm (GSA) [28] taking inspiration from law of gravity and interaction between the masses, Harmony Search Algorithm (HSA) [12] inspired by improvisation done by jazz musician, Differential Evolution (DE) algorithm [30] inspired by theory of evolution and Spider Monkey Optimization (SMO) [4] algorithm taking inspiration from foraging behaviour of spider monkeys.

Recently various variants of ABC algorithm have been proposed which includes modified global best artificial bee colony for constrained optimization problem [3], artificial bee colony algorithm with multiple search strategies [11], an adaptive artificial bee colony algorithm for global optimization [35], hybrid artificial bee colony with differential evolution [18][34], simulated annealing based artificial bee colony algorithm for global numerical optimization [6] and escalated convergent artificial bee colony [17]. Study has shown that these algorithms are considered as an efficient solver of complex optimization problems. Artificial Bee Colony (ABC) optimization algorithm and its variants has been applied to various optimization problems such as solving partition and scheduling problem in codesign [24][14], artificial neural networks [25], forecasting stock markets [15], automatic software fault localization [16], parameter identification for Van Der Pol - Duffing oscillator [10], network topology design [29] and structural engineering [8].

Since the inception of ABC algorithm, its different characteristics have been analyzed and investigated, numerically. However, a little work has been done in the theoretical analysis of this class of algorithms.

Jagdish Chand Bansal is with the Department of Mathematics, South Asian University, New Delhi, India, e-mail: (jcbansal@sau.ac.in).

Anshul Gopal is with Department of Mathematics, South Asian University, India and Atulya K. Nagar is with Department of Mathematics and Computer Science, Faculty of Science, Liverpool Hope University, Liverpool, L16 9JD, UK.

Manuscript received October , 2018; revised .

Study of stability and convergence behavior of Particle Swarm Optimization (PSO) algorithm is carried out using Z-transformation [19][26][33]. Stability analysis of Differential Evolution (DE) algorithm [7][13], Bacterial Foraging Optimization (BFO) algorithm [5], Ant Colony Optimization (ACO) algorithm [1] and Gravitational Search algorithm (GSA) [9] is done using von Neumann stability criteria and Lyapunov's stability theorem. Recently, authors' have performed stability analysis of Artificial Bee Colony (ABC) optimization algorithm using von Neumann stability criteria for two- level finite difference scheme [2]. One of the important aspects is to find the conditions under which the algorithm converges and to study the propagation of potential solutions in the search space using position update equation of the algorithm. Hence, this analysis plays a significant role in the theoretical study of the algorithm. Upto authors' knowledge no attempt has yet been made for such kind of analysis of ABC algorithm.
In this paper, we investigate the behaviour of ABC algorithm within the convergent boundaries of the parameters. The consistency of finite difference scheme corresponding to position update equation of ABC algorithm with generated partial differential equation is investigated. Trajectory of potential solutions is proposed by investigating the obtained partial differential equation. Then inferences are made by observing the obtained partial differential equation and results obtained from numerical experiments.
Rest of the paper is organized as follows: on the backdrop of original ABC algorithm in Section II, the paper is followed by analysis of trajectory of potential solutions in Section III. Convergence analysis of ABC algorithm is performed in Section IV. Numerical experiments are carried out in Section V, outcomes are discussed in Section VI and findings are concluded in Section VII.

II. Artificial Bee Colony (ABC) Optimization Algorithm

Artificial Bee Colony (ABC) optimization algorithm [21] is a population-based optimization algorithm and makes use of iterative method in order to reach global optima. It consists mainly of four phases. After initialization, exploitation of search space is done employed and onlooker bee phase while exploration is done in scout bee phase. In Artificial Bee Colony (ABC) optimization algorithm employed bees and onlooker bees are present in equal number.
The ABC algorithm has four phases; initialization, employed bee, onlooker bee and scout bee [2].

A. Intialization

Artificial Bee Colony (ABC) optimization algorithm initiates with randomly generated potential solutions. For employed bees initial solutions are generated by using the equation:

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

where, $x_{i,j}$ represents j^{th} dimension of the i^{th} employed bee. x_j^{max} is the upper bound and x_j^{min} represents the lower bound of the j^{th} parameter respectively. μ represents uniform random number in the interval $[0, 1]$. N represents the swarm size and D is the dimension of considered problem. Also, in this phase of the algorithm abandonment counter (AC) is reset for each employed bee.

B. Employeed Bee Phase

A new candidate solution is generated corresponding to each employed bee in this phase. First, the solution of the employed bee is copied to new candidate solution $(v_i = x_i)$. Then, a randomly selected parameter 'j' of the solution is updated by using the equation:

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{r,j}) \tag{2}$$

where $i, r \in \{1, 2, 3..., N\}$, $j \in \{1, 2, 3..., D\}$ $and\, i \neq r$. x_r is randomly selected solution in the neighbourhood of x_i candidate. ϕ is the random number in $[-1, 1]$. Then fitness of the candidate solution is calculated as follows:

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

where fit_i is the fitness value and g_i is the value of objective function for i^{th} candidate solution. If better fitness value is achieved then the candidate solution replaces the current solution and the abandonment counter (AC) is reset to zero, otherwise incremented by one.

C. Onlooker Bee Phase

In order to get better solution each employed bee is selected by onlooker bee. The probability for the selection of i^{th} employed bee is calculated by:

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

where p_i is the probability of selection of i^{th} employed bee. The updation of selected solution is done by using equation (2). Again if better fitness is achieved then employed bee is replaced with onlooker bee and AC is reset to zero. Otherwise AC is increased by unity.

D. Scout Bee Phase

In this phase AC of all potential solutions are checked for predefined limit.Those employed bees whose AC has reached the predefined limit becomes scout bee. Thereafter for scout bees the solution is generated by using equation (1) and AC is set to zero. The scout bee then becomes employed bee and hence prevents stagnation of the algorithm.

III. Trajectory of potential solutions in Artificial Bee Colony (ABC) algorithm

A. Motivation

Nature inspired optimization algorithms are used to solve real-world optimization problems. It is not always necessary that the potential solutions converge to equilibrium point as the iteration increases, they may diverge from the desired equilibrium point.Therefore it is important to study the conditions under which the potential solutions and hence the ABC algorithm converge to the desired equilibrium point. ϕ and ψ are two parameters used in the position update equation of ABC algorithm. The study of convergence will provide necessary recommendations for parameter setting of the parameters ϕ and ψ. Also, it is necessary to understand how the solutions move near optimal solution in the search space. This motivates the authors to study the iteration-wise movement of solutions, which are updated using position update equation (2) of the ABC algorithm. Convergent region is defined as the region bounded by parameters ϕ and ψ, where the algorithm converges to equilibrium point.

B. Partial Differential Equation associated with position update equation of ABC algorithm

In this section, we generate initial value problem associated with position update equation of ABC algorithm with the help of finite difference scheme corresponding to position update equation given in equation (2).

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{r,j})$$

where $i, r \in \{1, 2, 3..., N\}$; $j \in \{1, 2, 3..., D\}$ and $i \neq r$. If t represents the present iteration and the solution got updated, then $x_{i,j}$ represents the solution at iteration t and $v_{i,j}$ represents the solution at iteration $(t+1)$. So the above equation can be written as [2]:

$$x_{i,j}(t+1) = x_{i,j}(t) \ + \ \phi(x_{i,j}(t) - x_{r,j}(t)) \tag{4}$$

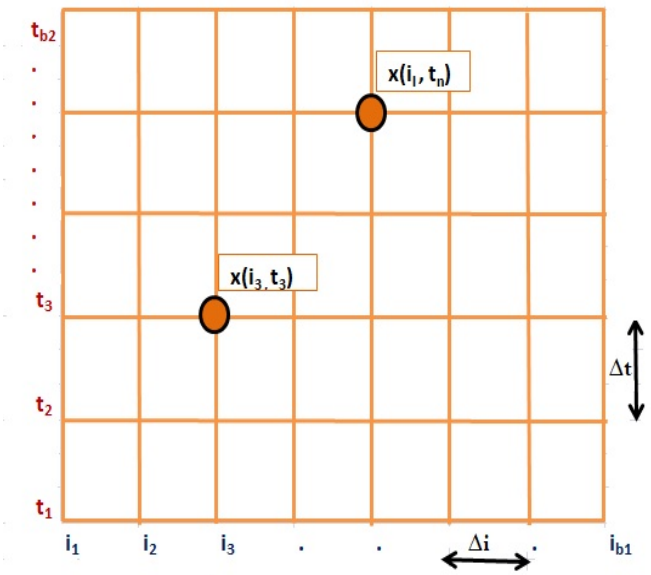

Fig. 1: Grid point representation of approximate solutions.

In ABC algorithm, the position update equation (4) is implemented component wise, i.e. the dimension of a solution is updated independently. The only link between the dimensions of the problem space is introduced via objective function. Thus, without loss of generality, for analysis purpose the algorithm description can be reduced to the one dimension case as considered in [33][9][7]. Thus equation (4) can be written as:

$$x_i(t+1) - x_i(t) = \phi(x_i(t) - x_r(t)) \tag{5}$$

where, r is randomly selected solution index different from i. If $x = x(i,t)$ represents true solution to a problem in an i-t computational domain. Then, $x_{l,n} = x(i_l, t_n)$ represents the approximate solution on the nodes of a uniform computational grid such that Δi=Δt, $l \in \{1, 2, 3..., b_1\}$ and $n \in \{1, 2, 3..., b_2\}$, where Δi and Δt are grid spacing in the direction of i and t respectively (shown in Figure 1).

In terms of grid points the difference equation can further be written as:

$$x(i_l, t_{n+1}) - x(i_l, t_n) = \phi(x(i_l, t_n) - x(i_r, t_n));\ x(i_l, 1) = g(i_l) \tag{6}$$

where, $l, r \in \{1, 2, 3..., b_1\}$ and $n \in \{1, 2, 3..., b_2\}$.
Since $l, r \in \{1, 2, 3..., b_1\}$, we can write $r = l + a$ where, $a \in Z/\{0\}$ such that $r \in \{1, 2, 3..., b_1\}$.
The above equation (6) can now be written as:

$$x(i_l, t_{n+1}) - x(i_l, t_n) = \phi(x(i_l, t_n) - x(i_{l+a}, t_n));\ x(i_l, 1) = g(i_l) \tag{7}$$

As discussed earlier since we have considered uniform grid for our analysis, so the grid spacings are of equal length i.e. Δt = Δi. Hence, equation (7) can be written as:

$$\frac{x(i_l, t_{n+1}) - x(i_l, t_n)}{\Delta t} = \frac{\phi(x(i_l, t_n) - x(i_{l+a}, t_n))}{\Delta i}; \tag{8}$$

$x(i_l, 1) = g(i_l)$
Since $a \in Z/\{0\}$, so the above equation (8) can be written as:

$$\frac{x(i_l, t_{n+1}) - x(i_l, t_n)}{\Delta t} = \frac{a\phi(x(i_l, t_n) - x(i_{l+a}, t_n))}{a\Delta i}; \tag{9}$$

$x_l^1 = x(i_l, 1) = g(i_l)$
or

$$\frac{x_l^{n+1} - x_l^n}{\Delta t} = \frac{a\phi(x_l^n - x_{l+a}^n)}{a\Delta i}; \tag{10}$$

$$x_l^1 = x(i_l, 1) = g(i_l)$$

where, $x_l^n = x(i_l, t_n)$. The partial differential equation corresponding to above difference equation and hence corresponding to position update equation of ABC algorithm with initial condition is given by:

$$\frac{\partial x}{\partial t} = -a\phi\frac{\partial x}{\partial i}; \quad x(i,1) = g(i) \tag{11}$$

or

$$\frac{\partial x}{\partial t} + v\frac{\partial x}{\partial i} = 0; \quad x(i,1) = g(i) \tag{12}$$

where, $\frac{\partial x}{\partial t} = \frac{x(i_l, t_{n+1}) - x(i_l, t_n)}{\Delta t}$, $\frac{\partial x}{\partial i} = \frac{(x(i_{l+a}, t_n) - x(i_l, t_n))}{a\Delta i}$, v= $a\phi$ and x is the exact solution of the partial differential equation (12).
By analyzing equation (12) we can infer that the partial differential equation corresponding to position update equation of ABC algorithm resembles 1- Dimension advection equation [20].
In the next subsection, we will verify that the difference scheme corresponding to position update equation of ABC algorithm given in equation (10) is consistent with the partial differential equation obtained in equation (12).

C. Consistency

Definition: [32] Let the partial differential equation under consideration be denoted by $Ly = F$ and the corresponding finite difference approximation by $L_l^n x_l^n = G_l^n$, where G_l^n denotes whatever approximation has been made of the source term. If we write the difference scheme as:

$$x_l^{n+1} = Qx_l^n + \Delta t G_l^n \tag{13}$$

where, $x_l^n = (..., x_{-1}^n, x_0^n, x_1^n ...)^T$, $G_l^n = (..., G_{-1}^n, G_0^n, G_1^n ...)^T$ and Q is an operator acting on the appropriate space. The difference scheme (13) is pointwise consistent with the partial differential equation $(Ly = F)$ if the solution of the partial differential equation, y, satisfies

$$y_l^{n+1} = Qy_l^n + \Delta t G_l^n + \Delta t \tau_l^n \tag{14}$$

and $\tau_l^n \to 0$ as $\Delta i, \Delta t \to 0$. We refer to τ_l^n as truncation error.
To show that the difference scheme (10) is consistent, we must write the scheme in the form of equation (13) as:

$$x_l^{n+1} = x_l^n + \phi\frac{a\Delta t}{a\Delta i}(x_l^n - x_{l+a}^n) \tag{15}$$

Equation (15) gives each component of an equation in the form of (13). Hence, to apply definition of consistency, we let x to be a solution of partial differential equation (12) and write:

$$\Delta t \tau_l^n = x_l^{n+1} - [x_l^n + \phi\frac{a\Delta t}{a\Delta i}(x_l^n - x_{l+a}^n)] \tag{16}$$

By expanding the components with the help of Taylor series we get:

$$\Delta t \tau_l^n = \Big[x_l^n + (x_t)_l^n \Delta t + (x_{tt})_l^n \frac{\Delta t^2}{2} + ...\Big] - \Big\{x_l^n + \phi\frac{a\Delta t}{a\Delta i}\Big[x_l^n - \Big(x_l^n + (x_i)_l^n a\Delta i + (x_{ii})_l^n \frac{(a\Delta i)^2}{2} - ..\Big)\Big]\Big\} \tag{17}$$

With proper rearrangement and cancellation of components in the above equation (17) we get:

$$\Delta t\tau_l^n = (x_t)_l^n \Delta t + a\phi(x_i)_l^n \Delta t + (x_{tt})_l^n \frac{\Delta t^2}{2} + \phi(x_{ii})_l^n \frac{a^2 \Delta t \Delta i}{2} + ... \tag{18}$$

By dividing both sides of equation (18) by Δt, we get the required equation as:

$$\tau_l^n = (x_t + a\phi x_i)_l^n + (x_{tt})_l^n \frac{\Delta t}{2} + \phi(x_{ii})_l^n \frac{a^2 \Delta i}{2} + ... \tag{19}$$

or

$$\tau_l^n = (x_t + vx_i)_l^n + (x_{tt})_l^n \frac{\Delta t}{2} + \phi(x_{ii})_l^n \frac{a^2 \Delta i}{2} + ... \tag{20}$$

Using equation (12) the above equation is modified to get the value of τ_l^n as:

$$\tau_l^n = (x_{tt})_l^n \frac{\Delta t}{2} + \phi(x_{ii})_l^n \frac{a^2 \Delta i}{2} + ... \tag{21}$$

or

$$\tau_l^n = o(\Delta t) + o(\Delta i) \tag{22}$$

According to the definition of consistency discussed previously, $\tau_l^n \to 0$ as $\Delta i, \Delta t \to 0$. Also, under the assumption that higher order derivatives of x are bounded at $(l\Delta i, n\Delta t)$ the considered difference scheme has accuracy of order $(1, 1)$.
Hence, the finite difference scheme given by equation (10) corresponding to position update equation of ABC algorithm is consistent with the obtained partial differential equation given by equation (12). In the next section, numerical experiments are done for deterministic analysis of the findings.

D. Lax Equivalence Theorem

Definition: A consistent, two level finite difference scheme for a well posed linear initial value problem is convergent iff it is stable [32].
Authors have already proved stability of position update equation of ABC algorithm in [2]. Also, consistency of the position update equation is proved in section 3.3. Since the corresponding initial value problem depends continuously upon its initial conditions, it is well posed too. Therefore by Lax equivalence theorem considered difference scheme corresponding to position update equation of ABC algorithm is convergent to the obtained partial differential equation given by equation (12).
Physical meaning of convergence:
$\Delta t \to 0$ indicates that distance between grid points in the direction of t tends to zero i.e., number of iterations reaches to maximum limit. While $\Delta i \to 0$ implies that, distance between grid points in the direction of i tends to zero i.e., distance between particles tend to zero, which means particles are getting clustered at a particular location in the search space.
These two inferences imply that when particles get clustered in the search space as number of iterations reaches to maximum limit, then the solutions obtained from difference equation (5) coincides with the solutions obtained from partial differential equation (12).

IV. Convergence Analysis of ABC Algorithm

In this section, the conditions under which the potential solutions considered in ABC algorithm converges to an equilibrium point is obtained. In order to do so, we first introduce a parameter ψ (as coefficient) in the position update equation of ABC algorithm.

$$v_{i,j} = \psi x_{i,j} + \phi(x_{i,j} - x_{r,j}) \,, \tag{23}$$

where $i, r \in \{1, 2, 3..., N\}$, $j \in \{1, 2, 3..., D\}$ $and\, i \neq r$. All the parameters are same as defined in section III. In case of original ABC algorithm the coefficient ψ is taken as unity. As explained in section

III, without loss of generality for analysis purpose the algorithm description can be reduced to one dimension case as considered in [33][9][7]. Thus equation (23) can be written as (as done in section **??**)

$$x_i(t+1) = \psi x_i(t) \ + \ \phi(x_i(t) - x_r(t)) \tag{24}$$

or

$$x_i(t+1) = (\psi + \phi)x_i(t) \ - \ \phi x_r(t) \tag{25}$$

For theoretical analysis of ABC algorithm, the deterministic version of position update equation is considered. Deterministic version is obtained by replacing random entity with the expected value. In equation (25), x_r is the position of randomly selected candidate solution in the neighbourhood of x_i. So, in order to obtain deterministic version of position update equation of ABC algorithm, we have to replace the position of randomly selected candidate solution by any expected value say 'p', i.e. $x_r = p$. Now equation (25) can be written as

$$x_i(t+1) = (\psi + \phi)x_i(t) \ - \ \phi p \tag{26}$$

The dynamic behaviour of potential solutions in the search space can be analysed by using the study of linear discrete-time dynamical system [33]. Equation (26) can be written in matrix form as

$$Y_{t+1} = AY_t + Bp \tag{27}$$

where, $Y_t = [x_i(t)]_{1*1}$, $A = [\psi + \phi]_{1*1}$ and $B = [-\phi]_{1*1}$
In accordance with the theory of dynamical systems, Y_t represents the particle state, comprising of its current position $x_i(t)$. The properties of dynamic matrix 'A' determines the time behaviour of the particle. External input 'p' helps in moving the particle towards specified position while input matrix 'B' provides the effect of external input on the particle state.

A. Equilibrium point

Definition: An equilibrium point is a state maintained by the dynamical system in the absence of external excitation (i.e., p= constant) [33].
If any equilibrium point exists then it satisfies the condition

$$Y_{t+1}^{eq} = Y_t^{eq} \ ; \qquad \forall \ t \tag{28}$$

By using the above condition in equation (26), the equilibrium point is calculated as:

$$x_i^{eq} = \frac{-\phi p}{1 - \psi - \phi} \tag{29}$$

Without loss of generality, equation (23) can also be simplified and written as

$$x_i(t+1) = \psi x_i(t) \ + \ \phi(x_r(t) - x_i(t)) \tag{30}$$

or

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

or

$$x_i(t+1) = (\psi - \phi)x_i(t) \ + \ \phi p \tag{32}$$

So, the corresponding equilibrium point is given by:

$$x_i^{eq} = \frac{\phi p}{1 - \psi + \phi} \tag{33}$$

For $\psi = 1$ i.e. in case of original ABC algorithm, the equilibrium point is calculated as $x_i^{eq} = p$ for both the cases.
It can be clearly observed that in the presence of coefficient ψ, the equilibrium point depends upon both parameters ψ and ϕ, whereas when coefficient $\psi = 1$ the equilibrium point is independent of both the parameters. Hence we can say that with the introduction of coefficient ψ ($\psi \neq 1$), the role of parameter ϕ became more significant in finding the equilibrium point as evident from equations (29) and (33).

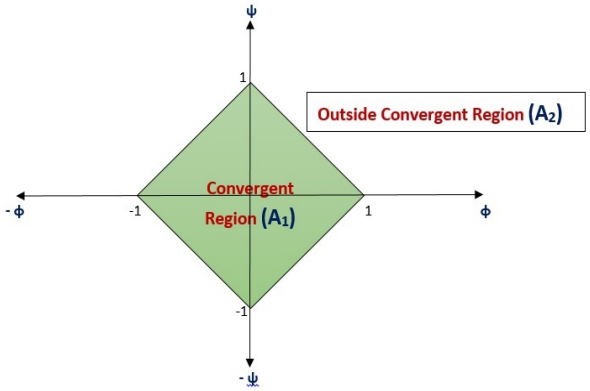

Fig. 2: Convergence boundary for parameters ψ and ϕ.

B. Convergence

In general, initially the particle state is not at equilibrium. So it is necessary to analyse whether the particle will eventually move towards equilibrium or not i.e., the optimization algorithm will converge or not. From the results of theory of dynamical system it can be concluded that the eigenvalues of the dynamic matrix A plays an important role in explaining the time behaviour of the potential solutions [22]. The necessary and sufficient condition for equilibrium point to be stable is that the magnitude of eigen values of the matrix A should be less than unity [33]. In this case the potential solutions will eventually settle at equilibrium and the algorithm will converge. By using equation (27), the matrix A has only one eigen value given by $\lambda_1 = \psi + \phi$. As discussed above for the convergence of ABC algorithm

$$|\lambda_1| < 1 \quad i.e. \quad |\psi + \phi| < 1 \tag{34}$$

Without loss of generality, equation (23) can also be written as:

$$x_i(t+1) = \psi x_i(t) \; + \; \phi(x_r(t) - x_i(t)) \tag{35}$$

or

$$x_i(t+1) = (\psi - \phi) x_i(t) \; + \; \phi x_r(t) \tag{36}$$

The corresponding eigen value will be given by $\lambda_2 = \psi - \phi$ and hence the condition for convergence of ABC algorithm will be given by:

$$|\lambda_2| < 1 \quad i.e. \quad |\psi - \phi| < 1 \tag{37}$$

By combining results from equation (34) and (37) the condition for convergence of ABC algorithm with coefficient ψ is:

$$|\psi + \phi| < 1 \quad and \quad |\psi - \phi| < 1 \tag{38}$$

The convergence boundary for the parameters ψ and ϕ is shown in Figure 2. In Figure 2, the shaded region (A_1) is termed as convergent region and (A_2) as outside convergent region. Convergent region signifies that, if the values of parameters ϕ and ψ lie within region (A_1) then the potential solutions will converge to equilibrium point. In next section, obtained conditions are verified numerically by testing them on benchmark test problems.

TABLE I: List of Test Problems (**AE**: Acceptable Error, **U**: Uni-modal, **M**: Multi-modal, **S**: Separable, **N**: Non-separable) [27] [31]

Name of the problem	Search Range	Optimum Value	Dim (n)	AE	Characteristic
Ackley	$[-30, 30]$	$f(\vec{0}) = 0$	30	$1.0E-05$	M, S
Alpine	$[-10, 10]$	$f(\vec{0}) = 0$	30	$1.0E-05$	M, S
Salomon	$[-100, 100]$	$f(\vec{0}) = 0$	30	$2.0E-01$	M, N
Pathological	$[-100, 100]$	$f(\vec{0}) = 0$	30	$1.05E-01$	M, N
Inverted Cosine wave	$[-5, 5]$	$f(\vec{0}) = -n+1$	10	$1.0E-05$	M, S
Shifted Rosenbrock	$[-100, 100]$	$f(o) = f(\vec{0}) = 390$	10	$1.0E-01$	M, N
De Jong	$[-5.12, 5.12]$	$f(\vec{0}) = 0$	30	$1.0E-05$	U, S
Moved axis parallel hyper-ellipsoid	$[-5.12, 5.12]$	$f(\vec{0}) = 0$	30	$1.0E-15$	U, S
Shifted Rotated and expanded uneven minima	$[-100, 100]$	$f(\vec{0}) = 500$	30	1913.0	M, N
Composition function 3	$[-100, 100]$	$f(\vec{0}) = 1100$	30	325.0	M, N

V. Numerical Experiment

In order to justify our theoretical findings for convergence analysis of ABC algorithm, numerical testing is carried out on a set of 10 benchmark problems. The benchmark problems considered have objective function with properties which includes unimodal, multimodal, separable and non-separable functions. In addition to that few have optimal point at origin and others have optimal point away from origin as explained in Table I. While doing numerical experiments to verify the findings of convergence analysis, two cases are considered. Firstly, when parameters ϕ and ψ lie within convergence region and secondly when they lie outside convergence region.

Following three types of numerical experiments are performed:

1) In subsection (V-A), movement of potential solution in the search space is analysed.
2) In subsection (V-B), efficiency of the algorithm is tested by calculating average number of function evaluations.
3) In subsection (V-C), accuracy of the algorithm is checked by analysing the mean error of considered test problems.

A. Movement of potential solution in the search space

In order to justify our finding that movement of solutions in ABC algorithm follows advection equation, numerical analysis is done with usual parameter settings. The general solution of the partial differential equation (11) can be written as **[Appendix A]**:

$$x(i,t) = g(i - a\phi(t-1)) \tag{39}$$

While, the position update equation of ABC algorithm along with initial solution $x_i(1)$ is given by:

$$x_i(t+1) = x_i(t) \; + \; \phi(x_i(t) - x_r(t)); \quad x_i(1) = g(i) \tag{40}$$

Following parameter settings are considered for the numerical experiment:

1) Since in the position update equation of ABC algorithm, the updated position depends upon the previous solution and a randomly selected solution, therefore for analysis purpose, minimum number of required solutions are two. So, we will consider a swarm size of 2 for the numerical analysis.

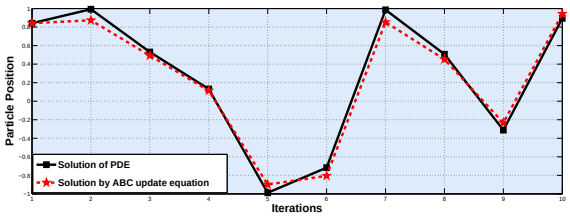

Fig. 3: Positions of solutions obtained from ABC update equation and solution of PDE.

TABLE II: Comparision of numerical results with random initialization **(i.e.,** $g(i) = 0.84147$)" (**PDE**: Partial differential equation)

S.No.	Value of parameter ϕ	Position of random particle	Position obtained from ABC update equation	Position obtained from solution of PDE	Absolute Error
1	0.50	(Initialization)	0.84147	0.84147	0
2	-0.45	0.77027	0.87351	0.99271	0.11920
3	0.22	-0.85826	0.49252	0.53118	0.03866
4	-0.67	1.05796	0.11367	0.13121	0.01754
5	-0.89	1.25168	-0.89916	-0.98841	0.08925
6	0.36	-0.63818	-0.80521	-0.71735	0.08786
7	0.98	0.88709	0.85325	0.98598	0.13273
8	-0.23	2.59738	0.45210	0.50690	0.05480
9	0.95	-0.26859	-0.23256	-0.31154	0.07898
10	-0.71	-1.88736	0.94235	0.89427	0.04808

2) Since we are performing numerical analysis and swarm size is fixed to two solutions, so the random solution index (r) will be taken as r = 2, ($r \neq i$).
3) In order to follow the phenomena of exploration and exploitation, some random values of parameter ϕ are considered in the range [-1, 1], given in Table II.
4) The concept of greedy selection is used while choosing the position of a random solution in the neighbourhood of the candidate solution.
5) Initialization function $g(i)$ is selected randomly (given in Table II) for performing the numerical analysis.

Now we will compare the numerical results obtained by solving equation (39) and (40), simultaneously the outcome of the numerical results is listed in Table II and shown graphically in Figure 3.

B. Efficiency of the algorithm

In order to check the efficiency of ABC algorithm, numerical experiments are performed when parameters are considered within and outside convergent region. Following parameter settings are considered while doing the numerical experiment.

1) Swarm size: 50
2) Maximum number of runs: 51
3) Maximum number of iterations: 6000
4) Acceptable error: Table I

The average number of function evaluations (AFEs) are calculated and reported in Table III for both the cases, i.e. when parameters lie within the convergent region A_1 (as shown in Figure 2) and outside convergent region A_2 (shown in Figure 2). The algorithm is stopped when either acceptable error is reached

TABLE III: Average number of Function Evaluations (AFEs) for region A_1 and A_2 (see Figure 2) (**TP**: Test Problem, A_1: Parameters within convergence region, A_2: Parameters outside convergence region)

TP	**AFEs for** A_1	**AFEs for** A_2
f_1	**22698**	198610
f_2	**27900**	200026
f_3	**198305**	200030
f_4	**156353**	200028
f_5	**14538**	199371
f_6	75	75
f_7	**5022**	27805
f_8	**28573**	200025
f_9	**261833**	300050
f_{10}	**168729**	299221

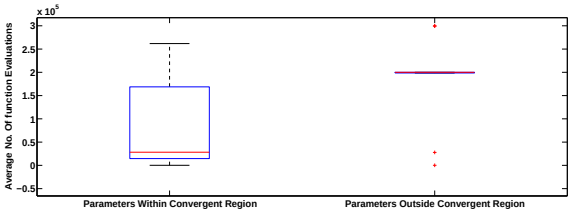

Fig. 4: Boxplot comparison of AFEs for region A_1 and A_2 (A_1: Parameters within convergent region, A_2: Parameters outside convergent region).

or maximum number of function evaluation is attained whichever is earlier. From the experimental results it is clear that ABC algorithm is efficient when parameters lie within A_1. The difference of results when parameters lie in A_1 and A_2 can be seen from the boxplot shown in Figure 4.

C. Accuracy of the algorithm

To check the accuracy of ABC algorithm, numerical experiments have been carried out by considering parameters within and outside convergent region. Parameter setting and test problems are same as considered in section V-B. Mean Error (ME) is calculated for considered test problems and numerical results are presented in Table IV. The improvement of accuracy of ABC algorithm with parameters within region A_1 and A_2 can be observed from the boxplot shown in Figure 5. Numerical results are again verified by performing non parametric test namely, Wilcoxon signed rank test and presented in Table V. If the obtained data sets have significant difference then we say that null hypothesis is rejected and '+' sign appears otherwise null hypothesis is accepted and '=' sign appears. In Table V, '+' sign appears 9 times out of 10. Thus the accuracy of ABC algorithm is better when parameters ϕ and ψ are considered in convergent region A_1.

VI. Discussion

The above theoretical and numerical results of convergence analysis of ABC algorithm reveals that the algorithm performs efficiently and accurately when parameters ϕ and ψ lie within convergent region. Presence of coefficient ψ ($\psi \neq 1$) in the position update equation of ABC algorithm makes the equilibrium point to be dependent upon both the parameters ϕ and ψ. Whereas, when parameter $\psi = 1$ (i.e., in case of original ABC algorithm) the equilibrium point becomes independent of both the parameters as explained in

TABLE IV: Mean Error (ME) for region A_1 and A_2 (**TP**: Test Problem, A_1: Parameters within convergent region, A_2: Parameters outside convergent region)

TP	**ME for A_1**	**ME for A_2**
f_1	**8.42E-06**	2.72E-05
f_2	**7.48E-06**	0.127836
f_3	**0.57**	1.61
f_4	**0.097**	13.837
f_5	**6.17E-06**	0.348546
f_6	**8.32E05**	1.04E06
f_7	**4.47E-06**	4.72E-06
f_8	**8.6E-16**	2.04E-13
f_9	**2117.71**	2509.26
f_{10}	**325.47**	360.76

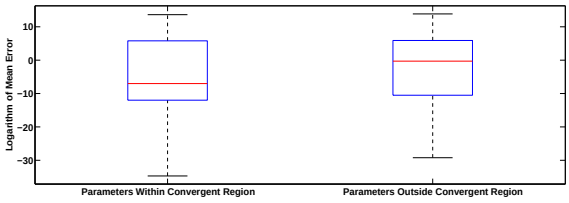

Fig. 5: Boxplot comparison of ME for region A_1 and A_2 (A_1: Parameters within convergent region, A_2: Parameters outside convergent region).

TABLE V: Comparision of Mean Error (ME) for region A_1 and A_2 using Wilcoxon Sign Rank test(**TP**: Test Problem, A_1: Parameters within convergent region, A_2: Parameters outside convergent region)

TP	**Based on ME for A_1 Vs A_2**
f_1	**+**
f_2	**+**
f_3	**+**
f_4	**+**
f_5	**+**
f_6	**+**
f_7	**=**
f_8	**+**
f_9	**+**
f_{10}	**+**
Number of + signs	**9**

section IV. From the numerical experiments (Table III) it can be easily seen that the AFEs are minimum for test problems in convergent region which shows that the test problems are converging towards equilibrium point in less number of iterations when parameters are in the convergent region. Results are further verified statistically by boxplot analysis of AFEs and ME. Again it can be seen that better results are obtained when parameters are within convergent region. The mean error for test problems in convergent region is also minimum as compared to the case when parameters lie outside convergent region which shows that the accuracy is better within convergent region. The results are verified by non parametric test, namely wilcoxon signed rank test. It can be inferred that result are significantly better when parameters are within convergent region.

Also, theoretical and numerical analyses of ABC position update equation reveals that the difference scheme (10) corresponding to position update equation (III-B) of ABC algorithm is consistent with the partial differential equation (12). The obtained partial differential equation infact resembles 1-Dimensional advection equation. The general solution of the partial differential equation obtained in equation (12) is given by $x(i,t)$=$f(i-vt)$. Where 'f' is an arbitrary function depending upon the initial condition and v= $a\phi$. The solution obtained describes an arbitrary shaped pulse which is swept along by the flow at constant speed 'v' without changing shape. Since $a \in Z/\{0\}$ and $\phi \in [-1,1]$, speed 'v' can be either positive or negative. Hence, the arbitrary shaped pulse generated can move in either forward direction or backward which explains that ABC algorithm can explore the entire search space. The analysis also explains a significant result that the propagation of particles depends upon the initial condition, i.e. the initialization phase of ABC algorithm plays an important role in the movement of solutions in the search space.

VII. Conclusion

Analysis of finding the conditions under which an algorithm converges to an equilibrium point plays a very vital role in making the algorithm efficient, reliable and accurate. For nature inspired algorithms, the solution update process depends upon the guided random search, which makes algorithm's nature probabilistic. This probabilistic nature of these algorithms makes the convergence analysis a difficult task. Convergence analysis of ABC algorithm is performed using the results from the theory of dynamical systems. Also condition of convergence of algorithm to equilibrium point is derived, which depends upon the parameters ϕ and ψ. Numerical experiments are performed to verify the findings and a convergent region is recommended for the values of ϕ and ψ. It can be concluded that the algorithm performs better when parameters are considered from the convergent region.

Also, study of the movement of solutions in the search space is important to analyse the search behaviour of ABC algorithm. In order to carry out this study, partial differential equation associated with position update equation of ABC algorithm is obtained. The obtained partial differential equation resembles 1-Dimensional advection equation. It was also proved that the finite difference scheme corresponding to position update equation of ABC algorithm is consistent with the obtained partial differential equation. So, from the general solution of the partial differential equation, it can be concluded that the solutions obtained by ABC algorithm propagate in an arbitrary shaped pulse which is swept along the flow at constant speed without changing shape.

Acknowledgment

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

Appendix A
General solution of PDE

The general solution of partial differential equation (12) is given by:

$$x(i,t) = f(i - a\phi t) \tag{41}$$

TABLE VI: List of Test Problems considered for numerical experiments [27] [31]

Name of the problem	Objective function
Ackley	$Minf_1(x) = -20 + e + exp(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n} {x_i}^3})$
Alpine	$Minf_2(x) = \sum_{i=1}^{n} \lvert x_i \sin x_i + (0.1)x_i \rvert$
Salomon	$Minf_3(x) = 1 - cos(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}) + 0.1(\sqrt{\sum_{i=1}^{n} x_i^2})$
Pathological	$Minf_4(x) = \sum_{i=1}^{n-1} \left(\frac{\sin^2\left(\sqrt{x_{i+1}^2+100x_i^2}\right)-0.5}{0.001(x_{i+1}^2-2x_{i+1}x_i+x_i^2)^2+1.0} + 0.5 \right)$
Inverted Cosine wave	$Minf_5(x) = -\sum_{i=1}^{n-1} \left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})}{8}\right) \times \mathrm{i} \right)$
Shifted Rosenbrock	$Minf_6(x) = \sum_{i=1}^{n-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}$, $z = x - o + 1$, $x = [x_1, x_2, ...x_n]$, $o = [o_1, o_2, ...o_n]$
De Jong	$Minf_7(x) = \sum_{i=1}^{n} i.(x_i)^4$
Moved axis parallel hyper-ellipsoid	$Minf_8(x) = \sum_{i=1}^{n} 5i \times x_i^2$
Shifted Rotated and expanded uneven minima	$Minf_9(x) = (CEC2015)$
Composition function 3	$Minf_{10}(x) = (CEC2015)$

with initial condition given as:

$$x(1,t) = g(i) \tag{42}$$

By using equation (43) and (44) we get

$$x(1,t) = f(i - a\phi) = g(i) \tag{43}$$

or

$$f(I) = g(I + a\phi), \quad where \;\; I = i - a\phi \tag{44}$$

By using equation (41) and (44) we get the final solution of partial differential equation as

$$x(i,t) = g(i - a\phi t + a\phi) \tag{45}$$

or

$$x(i,t) = g(i - a\phi(t-1)) \tag{46}$$

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] Jagdish Chand Bansal, Anshul Gopal, and Atulya K Nagar. Stability analysis of artificial bee colony optimization algorithm. *Swarm and Evolutionary Computation*, 2018.

[3] Jagdish Chand Bansal, Susheel Kumar Joshi, and Harish Sharma. Modified global best artificial bee colony for constrained optimization problems. *Computers & Electrical Engineering*, 2017.

[4] Jagdish Chand Bansal, Harish Sharma, Shimpi Singh Jadon, and Maurice Clerc. Spider monkey optimization algorithm for numerical optimization. *Memetic computing*, 6(1):31–47, 2014.

[5] 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.

[6] Shi-Ming Chen, Ali Sarosh, and Yun-Feng Dong. Simulated annealing based artificial bee colony algorithm for global numerical optimization. *Applied mathematics and computation*, 219(8):3575–3589, 2012.

[7] 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.

[8] 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.

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

[10] Fei Gao, Xue-jing Lee, Feng-xia Fei, Heng-qing Tong, Yi-bo Qi, Yan-fang Deng, Ilangko Balasingham, and Hua-ling Zhao. Parameter identification for van der pol–duffing oscillator by a novel artificial bee colony algorithm with differential evolution operators. *Applied Mathematics and Computation*, 222:132–144, 2013.

[11] Wei-feng Gao, Ling-ling Huang, San-yang Liu, Felix TS Chan, Cai Dai, and Xian Shan. Artificial bee colony algorithm with multiple search strategies. *Applied Mathematics and Computation*, 271:269–287, 2015.

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

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

[14] Shih-Cheng Horng. Combining artificial bee colony with ordinal optimization for stochastic economic lot scheduling problem. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 45(3):373–384, 2015.

[15] 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.

[16] Linzhi Huang and Jun Ai. Automatic software fault localization based on artificial bee colony. *Journal of Systems Engineering and Electronics*, 26(6):1325–1332, 2015.

[17] Shimpi Singh Jadon, Jagdish Chand Bansal, and Ritu Tiwari. Escalated convergent artificial bee colony. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(1-2):181–200, 2016.

[18] Shimpi Singh Jadon, Ritu Tiwari, Harish Sharma, and Jagdish Chand Bansal. Hybrid artificial bee colony algorithm with differential evolution. *Applied Soft Computing*, 58:11–24, 2017.

[19] 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.

[20] Takeo Kajishima and Kunihiko Taira. Finite-difference discretization of the advection-diffusion equation. In *Computational Fluid Dynamics*, pages 23–72. Springer, 2017.

[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] Anatole Katok and Boris Hasselblatt. *Introduction to the modern theory of dynamical systems*, volume 54. Cambridge university press, 1995.

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

[24] Mouloud Koudil, Karima Benatchba, Amina Tarabet, and El Batoul Sahraoui. Using artificial bees to solve partitioning and scheduling problems in codesign. *Applied Mathematics and Computation*, 186(2):1710–1722, 2007.

[25] Pravin Yallappa Kumbhar and Shoba Krishnan. Use of artificial bee colony (abc) algorithm in artificial neural network synthesis. *International Journal of Advanced Engineering Sciences and Technologies*, 11(1):162–171, 2011.

[26] 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.

[27] BY Qu, JJ Liang, PN Suganthan, and Q Chen. Problem definitions and evaluation criteria for

the cec 2015 competition on single objective multi-niche optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Tech. Rep.*, 2014.

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

[29] 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*, 38:187–201, 2018.

[30] 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.

[31] Ponnuthurai N Suganthan, Nikolaus Hansen, Jing J Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report*, 2005005:2005, 2005.

[32] James William Thomas. *Numerical partial differential equations: finite difference methods*, volume 22. Springer Science & Business Media, 2013.

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

[34] Jing Yang, Wen-Tao Li, Xiao-Wei Shi, Li Xin, and Jian-Feng Yu. A hybrid abc-de algorithm and its application for time-modulated arrays pattern synthesis. *IEEE Transactions on Antennas and Propagation*, 61(11):5485–5495, 2013.

[35] Alkın Yurtkuran and Erdal Emel. An adaptive artificial bee colony algorithm for global optimization. *Applied Mathematics and Computation*, 271:1004–1023, 2015.

Jagdish Chand Bansal has recieved his Ph.D. from Indian Institute of Technology Roorkee, India. He is currently an Assistant Professor at South Asian University, New Delhi, India and Visiting Research Fellow at Liverpool Hope University, UK. He is the editor-in-chief of International Journal of Swarm Intelligence (IJSI) published by Inderscience. He is also the editor-in-chief of Springer book series Algorithms for Intelligent Systems. His primary area of interest is nature-inspired optimization techniques.

Anshul Gopal received his M.Sc. degree in Mathematics from University of Delhi, New Delhi, India in 2011. Currently, he is perusing Ph.D. from South Asian University, New Delhi, India. His area of research is nature-inspired optimization algorithms and soft computing.

Atulya K. Nagar holds the Foundation Chair as Professor of Mathematical Sciences and is Dean of the Faculty of Science at Liverpool Hope University, UK. Prof. Nagar is an internationally respected scholar working at the cutting edge of theoretical computer science, applied mathematical analysis, operations research, and systems engineering.He received a prestigious Commonwealth Fellowship for pursuing his doctorate (DPhil) in Applied NonLinear Mathematics, which he earned from the University of York (UK) in 1996; and he holds BSc (Hons.), MSc, and MPhil (with Distinction) from the MDS University of Ajmer, India. He has edited volumes on Applied Mathematics, and Intelligent Systems and is the Editor-in-Chief of the International Journal of Artificial Intelligence and Soft Computing (IJAISC) and serves on editorial boards for a number of prestigious journals such as the Journal of Universal Computer science. He is well published with over 350 publications in prestigious publishing outlets such as the Journal of Applied Mathematics, the IEEE Transactions, Discrete Applied Mathematics, Fundamenta Informaticae to name a few.