Arabian Journal for Science and Engineering (2020) 45:2385–2394
https://doi.org/10.1007/s13369-019-03991-8

RESEARCH ARTICLE - SPECIAL ISSUE - INTELLIGENT COMPUTING AND INTERDISCIPLINARY APPLICATIONS

On Stability Analysis of Particle Swarm Optimization Algorithm

Anshul Gopal[1] · Mohammad Mahdi Sultani[1] · Jagdish Chand Bansal[1]

Received: 26 November 2018 / Accepted: 19 June 2019 / Published online: 3 July 2019

Abstract
Particle swarm optimization (PSO) is one of the most efficient and popular swarm intelligence-based search algorithms for continuous optimization. PSO provides the solutions probabilistically. Therefore, finding error bound during the search process can help in developing a better PSO. Stability analysis of an algorithm provides the information about error bounds. Stability analysis of PSO with inertia weight and constriction coefficient is carried out by von Neumann stability criterion. Conditions on acceleration parameters, constriction coefficient and inertia weight are obtained for stability.

Keywords Particle swarm optimization (PSO) algorithm · Stability analysis · Finite difference scheme · von Neumann stability criterion · Inertia weight · Constriction coefficient

1 Introduction

Recently, algorithms taking inspiration from social behaviour metaphor and natural phenomena have attracted researchers. This class of algorithms includes particle swarm optimization (PSO) algorithm [25], differential evolution (DE) algorithm [36], artificial bee colony (ABC) algorithm [24], gravitational search algorithm (GSA) [31], harmony search algorithm (HSA) [16], spider monkey optimization (SMO) algorithm [4], genetic algorithm (GA) [18], etc. These algorithms are efficient and effective solver of complex optimization problems.

PSO is a swarm intelligence-based search algorithm, inspired by birds' flocking or fish schooling. The social cooperative behaviour of birds in finding their food or nest inspired the PSO working mechanism. Working of PSO algorithm is explained in Sect. 2. In very short span of time, PSO has been modified in many ways and applied to many application problems. Chaotic PSO [26], multi-objective PSO algorithm [32], various hybrid PSO algorithms [12,17,29] and PSO for constrained optimization problems [21] are few contributions in the development of PSO. The PSO algorithm and its variants are applied to various discrete and continuous optimization problems from the field of neural networks [28,42], data clustering [40], optimal power flow [1], stock market prediction [20] and assignment problem [34].

Researchers have analysed these meta-heuristic algorithms experimentally and analytically, but little work has been done in theoretical study of this class of algorithms. Stability and convergence analysis of few algorithms have already been carried out which include particle swarm optimization (PSO) algorithm [7], artificial bee colony (ABC) algorithm by using von Neumann stability analysis in [2,3], differential evolution (DE) algorithm by using von Neumann and Lyapunov stability criterion in [11,19], gravitational search algorithm (GSA) by taking results from Lyapunov stability criterion [13] and bacterial foraging optimization (BFO) algorithm [5].

Theoretical analysis of stability and convergence behaviour of PSO algorithm has been previously analysed by using standard results from theory of dynamical system. The analysis recommends the parameter selection in PSO [39]. Stability and convergence properties of PSO algorithm are analysed in [10], leading to a generalized model of the algorithm in which convergence tendency of the system was controlled by a set of parameters. In [22], stochastic process theory was used for deriving stochastic convergent condition of the particle swarm system. Concept of passive system

✉ Jagdish Chand Bansal
jcbansal@sau.ac.in

Anshul Gopal
anshulgopal@students.sau.ac.in

Mohammad Mahdi Sultani
mahdisu53@gmail.com

[1] South Asian University, New Delhi, India

and Lyapunov stability analysis is used in [23] for deriving stability conditions of PSO algorithm. In [7], the concept of convergence and stability of linear recurrence relations is considered for deriving stability condition. The properties of stability and convergence of standard particle swarm optimizer (SPSO) 2011 were analysed in [6]. Also, identification of boundaries of parameters was done, which insures convergence of particle to their equilibrium point. The concept of upper border stability limit (USL) curve was introduced in [15], and it was shown that performance of PSO algorithm is better when parameter selection is done close to USL curve. Based on weak stagnation assumption, order-2 stability of PSO algorithm was analysed in [27]. In [37], convergence of a variant of PSO, namely quantum-behaved PSO (QPSO), to its global optima was analysed on a probabilistic metric space. Based on discrete-time dynamic system theory, convergence of particle to equilibrium point was analysed in [9]. Stochastic stability analysis of linear continuous and discrete PSO algorithm is done in [14]. In [30], theory of convergence of stochastic sequence is used in analysing convergence of PSO algorithm and parameter selection is proposed based on the analysis. In [8], convergence analysis of multi-objective PSO was done. Also, conditions on its parameters were proposed that guide the convergence of algorithm to the optimal Pareto front in the objective function space. The present work verifies the parameter tuning done for the stability of PSO algorithm with inertia weight and proposes stability condition for PSO algorithm with constriction coefficient by using von Neumann stability criterion for finite difference scheme.

In rest of the paper, Sect. 2 explains PSO algorithm, followed by motivation and von Neumann stability criterion in Sect. 3. Stability analysis of PSO algorithm with inertia weight (w) and constriction coefficient (χ) is done in Sect. 4. The obtained stability condition is verified by performing numerical experiments in Sect. 5, and findings are concluded in Sect. 6.

2 Particle Swarm Optimization (PSO) Algorithm

2.1 Standard Algorithm

Particle swarm optimization (PSO) algorithm was developed by Kennedy and Eberhart [25] in 1995 based on the social behaviour metaphor. The initialization of the algorithm is done randomly with candidate solutions, referred to as particles. The particles initially move in the problem space with randomized velocity assigned to them. The particles preferred to move towards the location with best fitness achieved by the particle itself and towards the best fitness location gained by the whole population so far. The inertial weight version of PSO algorithm is considered as standard particle swarm optimization. Therefore, for this study, consider inertia weight version of PSO [35]. In PSO algorithm, the particles update their positions based on the following two update equations:

$$\mathbf{v}_{d,t+1} = w\mathbf{v}_{d,t} + \mathbf{b}_1\mathbf{r}_1(\mathbf{p}_1 - \mathbf{x}_{d,t}) + \mathbf{b}_2\mathbf{r}_2(\mathbf{p}_2 - \mathbf{x}_{d,t}) \quad (1)$$

$$\mathbf{x}_{d,t+1} = \mathbf{x}_{d,t} + \mathbf{v}_{d,t+1} \quad (2)$$

Here, Eq. (1) is called velocity update equation and Eq. (2) is called position update equation. At tth iteration, the velocity $\mathbf{v}_{d,t}$ in dimension d is updated depending upon the weighted current velocity ($w\mathbf{v}_{d,t}$) and on the terms attracting the particles towards their own best position ($\mathbf{p}_1$) along with the best position of the whole population ($\mathbf{p}_2$). The coefficients ($\mathbf{b}_1$) and ($\mathbf{b}_2$) provide strength for attraction. The position of the particle is updated with the help of current position ($\mathbf{x}_{d,t}$) and the updated velocity ($\mathbf{v}_{d,t+1}$). The vector random numbers ($\mathbf{r}_1$) and ($\mathbf{r}_2$) provide useful randomness for better space exploration. They are generally taken as uniform random number between [0,1], i.e. $\mathbf{r}_1, \mathbf{r}_2 \in U[0, 1]$.

We can conclude from Eqs. (1) and (2) that the velocity and position updation is done independently for each dimension. Thus, without loss of generality, for analysis purpose the algorithm can be reduced to one dimension.

In order to further simplify the system and make it more understandable, we take p_1, p_2, r_1 and r_2 as constants for the remaining analysis [39]. Then, Eqs. (1) and (2) can be written as

$$v_{d,t+1} = wv_{d,t} + c_1(p_1 - x_{d,t}) + c_2(p_2 - x_{d,t}) \quad (3)$$

$$x_{d,t+1} = x_{d,t} + v_{d,t+1} \quad (4)$$

where $c_1 = b_1r_1$ and $c_2 = b_2r_2$
Making substitution in Eq. (4) from Eq. (3), we get the position update equation

$$x_{d,t+1} - (1 + w - c_1 - c_2)x_{d,t} + wx_{d,t-1} = c_1p_1 + c_2p_2 \quad (5)$$

The update Eq. (5) is a difference equation which will now be considered for stability analysis of PSO algorithm.

The next section presents the motivation for PSO's stability analysis. This section also presents the von Neumann stability criterion for a given finite difference scheme.

3 Motivation and von Neumann Stability Criterion

3.1 Motivation

Nature-inspired optimization algorithms make use of iterative procedure in order to solve real-world optimization problems and provide near-optimal solution corresponding to such problems. Since near-optimal solution is obtained, it leads to generation of error in subsequent iterations. So it is important to find conditions under which the error remains bounded depending upon various parameters present in the algorithm. Hence, parameter selection based on stability analysis plays a vital role in making algorithm efficient. Stability analysis is a deterministic analysis and nature-inspired optimization algorithms are stochastic in nature which makes such analysis of this class of algorithms more difficult. This motivates the authors to undergo stability analysis of PSO algorithm and to obtain suitable values of PSO parameters. von Neumann stability criterion is applied for the stability analysis of PSO. Next section explains von Neumann stability criterion.

3.2 von Neumann Stability Criterion

Consider a generalized linear partial differential equation represented by

$$\frac{\partial x}{\partial t} + M_d(x) = C$$

where $M_d(x)$ refers to a linear differential operator, C is constant and x is the dependent variable depending upon variables d and t.

Corresponding to considered linear partial differential equation, the generalized finite difference scheme is given by [41]

$$\sum_{p=-a_l}^{a_r} A_p x_{n,j+p} = \sum_{p=-b_l}^{b_r} B_p x_{n+1,j+p} + C \tag{6}$$

where a_l, a_r, b_l and b_r are non-negative integers. n and j represent number of grid points in the direction of d and t, respectively.

The von Neumann stability procedure consists of finding amplification factor (A) by firstly discretizing each term $x_{n,j}$ as x_{n_l,j_m}, where $l \in \{1, 2, 3 \ldots, b_1\}$ and $m \in \{1, 2, 3 \ldots, b_2\}$. Then, each discretized term of the difference equation is replaced by kth Fourier component of a harmonic decomposition of x_{n_l,j_m}, i.e. by taking $x_{n_l,j_m} = B_k e^{\iota\sigma_k j_m} e^{-\iota\beta_k n_l} = B_k e^{\iota\sigma_k m\Delta j} e^{-\iota\beta_k l\Delta n}$, where $\iota = \sqrt{-1}$, B_k represents the amplitude of kth component, β_k is the angular frequency and σ_k is the wave number of kth component [33]. The amplification factor (A) is given by $A = exp(-\iota\beta_k \Delta n)$.

Definition The necessary and sufficient condition for the stability of a finite difference scheme with only one dependent variable is that the modulus of amplification factor should be less than or equal to unity, i.e. $|A| \leq 1$. If $|A| = 1$, the finite difference scheme is said to be marginally stable and unstable when $|A| > 1$.

Stability is defined for a homogeneous finite difference scheme and the non-homogeneous part will contribute in the truncation term. Therefore, in order to discuss the stability of non-homogeneous difference equation, we consider the stability of associated homogeneous scheme [38].

In the next section, von Neumann criterion is applied to the finite difference equation corresponding to the update equation of PSO algorithm to get stability condition of PSO algorithm.

4 Stability Analysis

Theorem 1 *Particle swarm optimization algorithm with inertia weight w is said to be stable iff the acceleration coefficient c_1, c_2 and inertia weight w satisfies the condition,* $0 \leq (c_1 + c_2) \leq 2(1 + w)$.

Proof For stability analysis of PSO algorithm, consider the update Eq. (5) as finite difference scheme

$$x_{d,t+1} - (1 + w - c_1 - c_2)x_{d,t} + w x_{d,t-1} = c_1 p_1 + c_2 p_2$$

Equation (5) is a non-homogeneous finite difference scheme with $A_{-1} = w$, $A_0 = -(1 + w - c_1 - c_2)$, $A_1 = 1$, $B_{-1} = B_0 = B_1 = 0$ and $C = c_1 p_1 + c_2 p_2$. Since Eq. (5) is non-homogeneous finite difference equation, in order to find stability condition of PSO algorithm, for further analysis we will consider the following associated homogeneous difference scheme which is obtained by setting $c_1 p_1 + c_2 p_2 = 0$ in Eq. (5):

$$x_{d,t+1} - (1 + w - c_1 - c_2)x_{d,t} + w x_{d,t-1} = 0 \tag{7}$$

By using the transformation $t \to t + 1$ in Eq. (7), we get

$$x_{d,t+2} - (1 + w - c_1 - c_2)x_{d,t+1} + w x_{d,t} = 0 \tag{8}$$

or

$$x_{d,t+2} - \lambda x_{d,t+1} + w x_{d,t} = 0 \tag{9}$$

where $\lambda = (1+w-c_1-c_2)$. If the exact solution in d–t computational domain is taken as $x = x(d, t)$, the approximate solution at the nodes of the grid is given by $x(d_i, t_j)$, where $i \in \{1, 2, 3 \ldots, b_1\}$ and $j \in \{1, 2, 3 \ldots, b_2\}$ as shown in Fig. 1. Therefore, for stability analysis of PSO algorithm we consider finite difference scheme given by Eq. (9) instead of Eq. (5). The von Neumann stability criterion for finite difference scheme is used for deriving the stability condition for the update Eqs. (3) and (4) of PSO algorithm.

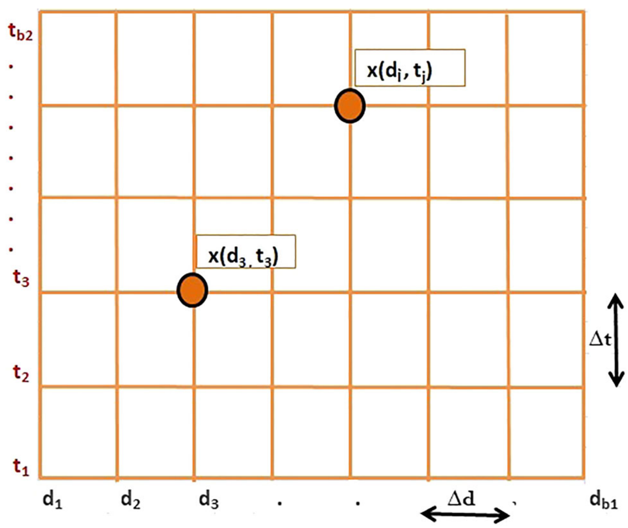

Fig. 1 Grid point representation of approximate solutions

Let the nth component of the complex Fourier series solution to the given equation is given by:

$$x(d_i, t_j) = B_n e^{\iota(\sigma_n d_i - \beta_n t_j)} \tag{10}$$

or

$$x_{d_i,t_j} = B_n e^{\iota(\sigma_n i\Delta d - \beta_n j\Delta t)} \tag{11}$$

where $\iota = \sqrt{-1}, d_i = i\Delta d, t_j = j\Delta t$, B_n represents the amplitude of nth component, β_n is the angular frequency and σ_n is the wave number of nth component [19].

In terms of grid point (d_i, t_j), Eq. (9) can be written as

$$x_{d_i,t_{j+2}} - \lambda x_{d_i,t_{j+1}} + w x_{d_i,t_j} = 0 \tag{12}$$

Substituting the value of x_{d_i,t_j} from Eq. (11) to Eq. (12), we get

$$B_n e^{\iota(\sigma_n i\Delta d - \beta_n j\Delta t)}(e^{-\iota\beta_n 2\Delta t} - \lambda e^{-\iota\beta_n \Delta t} + w) = 0 \tag{13}$$

Since $B_n \neq 0$ until the algorithm terminates,

$$e^{-\iota\beta_n 2\Delta t} - \lambda e^{-\iota\beta_n \Delta t} + w = 0 \tag{14}$$

or

$$A^2 - \lambda A + w = 0 \tag{15}$$

where $A = exp(-\iota\beta_n \Delta t)$ = amplification factor
Solving the quadratic Eq. (15), the amplification factor is obtained as

$$A = \frac{\lambda \pm \sqrt{\lambda^2 - 4w}}{2}$$

Now according to von Neumann stability criterion, the finite difference scheme (9) is stable iff for the amplification factor (A), $|A| \leq 1$ [41]. Therefore, we consider the finite difference scheme given by Eq. (9) and so the finite difference scheme given by Eq. (5), and hence, the PSO algorithm is stable iff $|A| \leq 1$.

$$i.e. \quad \left| \frac{\lambda \pm \sqrt{\lambda^2 - 4w}}{2} \right| \leq 1$$
$$\Rightarrow \qquad -1 \leq \frac{\lambda \pm \sqrt{\lambda^2 - 4w}}{2} \leq 1,$$
$$where \quad \lambda = (1 + w - c_1 - c_2)$$

Now the following two cases arise
Case 1:

$$\frac{\lambda \pm \sqrt{\lambda^2 - 4w}}{2} \leq 1 \tag{16}$$

or

$$\pm\sqrt{\lambda^2 - 4w} \leq 2 - \lambda$$
$$\Rightarrow \quad |\sqrt{\lambda^2 - 4w}| \leq 2 - \lambda \tag{17}$$

By squaring both the sides, we get

$$\lambda^2 - 4w \leq (2 - \lambda)^2 \tag{18}$$

Replacing λ by $(1 + w - c_1 - c_2)$ in the inequality given by Eq. (18), we get a stability condition

$$(c_1 + c_2) \geq 0 \tag{19}$$

Case 2:

$$-1 \leq \frac{\lambda \pm \sqrt{\lambda^2 - 4w}}{2} \tag{20}$$

or

$$-(2 + \lambda) \leq \pm\sqrt{\lambda^2 - 4w}$$
$$\Rightarrow \quad (2 + \lambda) \geq \pm\sqrt{\lambda^2 - 4w}$$
$$\Rightarrow \quad (2 + \lambda) \geq |\sqrt{\lambda^2 - 4w}| \tag{21}$$

By squaring both the sides, we get

$$(2 + \lambda)^2 \geq (\lambda^2 - 4w) \tag{22}$$

Replacing λ by $(1 + w - c_1 - c_2)$ in the inequality given by Eq. (22), we get another stability condition

$$(c_1 + c_2) \leq 2(1 + w) \tag{23}$$

From Eqs. (19) and (23), we get the condition for stability as

$$0 \le (c_1 + c_2) \le 2(1 + w) \tag{24}$$

Thus, the PSO algorithm is stable iff the acceleration coefficients and the inertial weight satisfy the inequality given by Eq. (24).

Theorem 2 *Particle swarm optimization algorithm with constriction coefficient χ is said to be stable iff the acceleration coefficients c_1, c_2 and constriction coefficient χ satisfy the condition,* $0 \le (c_1 + c_2) \le 2(1 + 1/\chi)$.

Proof In PSO algorithm, the velocity and position update equation with constriction coefficient χ is given by

$$v_{d,t+1} = \chi(v_{d,t} + b_1 r_1(p_1 - x_{d,t}) + b_2 r_2(p_2 - x_{d,t})) \tag{25}$$

$$x_{d,t+1} = x_{d,t} + v_{d,t+1} \tag{26}$$

or

$$v_{d,t+1} = \chi(v_{d,t} + c_1(p_1 - x_{d,t}) + c_2(p_2 - x_{d,t})) \tag{27}$$

$$x_{d,t+1} = x_{d,t} + v_{d,t+1} \tag{28}$$

where $c_1 = b_1 r_1$, $c_2 = b_2 r_2$, χ is the constriction coefficient and other variables are same as explained in Sect. 2. □

Making substitution in Eq. (28) from Eq. (27), we get the position update equation

$$x_{d,t+1} - (1 + \chi - \chi(c_1 + c_2))x_{d,t} + \chi x_{d,t-1} = \chi(c_1 p_1 + c_2 p_2) \tag{29}$$

The update Eq. (29) is a difference equation which will now be considered for stability analysis of PSO algorithm with constriction coefficient. Equation (29) is a non-homogeneous finite difference scheme with $A_{-1} = \chi$, $A_0 = -(1 + \chi - \chi(c_1 + c_2))$, $A_1 = 1$, $B_{-1} = B_0 = B_1 = 0$ and $C = \chi(c_1 p_1 + c_2 p_2)$. Since Eq. (29) is non-homogeneous finite difference equation, in order to find stability condition of PSO algorithm, for further analysis we will consider the following associated homogeneous difference scheme which is obtained by setting $\chi(c_1 p_1 + c_2 p_2) = 0$ in Eq. (29):

$$x_{d,t+1} - (1 + \chi - \chi(c_1 + c_2))x_{d,t} + \chi x_{d,t-1} = 0 \tag{30}$$

By using the transformation $t \to t + 1$ in Eq. (7), we get

$$x_{d,t+2} - (1 + \chi - \chi(c_1 + c_2))x_{d,t+1} + \chi x_{d,t} = 0 \tag{31}$$

or

$$x_{d,t+2} - \mu x_{d,t+1} + \chi x_{d,t} = 0 \tag{32}$$

where $\mu = (1 + \chi - \chi(c_1 + c_2))$. If the exact solution in d-t computational domain is taken as $x = x(d, t)$, the approximate solution at the nodes of the grid is given by $x(d_i, t_j)$, where $i \in \{1, 2, 3 \ldots, b_1\}$ and $j \in \{1, 2, 3 \ldots, b_2\}$ as shown in Fig. 1. Therefore, for stability analysis of PSO algorithm we consider finite difference scheme given by Eq. (32) instead of Eq. (29). The von Neumann stability criterion for finite difference scheme is used for deriving the stability condition for the update Eqs. (27) and (28) of PSO algorithm.

The nth component of the complex Fourier series solution to the given equation is given by:

$$x(d_i, t_j) = B_n e^{\iota(\sigma_n d_i - \beta_n t_j)} \tag{33}$$

or

$$x_{d_i,t_j} = B_n e^{\iota(\sigma_n i \Delta d - \beta_n j \Delta t)} \tag{34}$$

where $\iota = \sqrt{-1}$, $d_i = i\Delta d$, $t_j = j\Delta t$, B_n represents the amplitude of nth component, β_n is the angular frequency and σ_n is the wave number of nth component [19].

In terms of grid point (d_i, t_j), Eq. (32) can be written as

$$x_{d_i,t_{j+2}} - \mu x_{d_i,t_{j+1}} + \chi x_{d_i,t_j} = 0 \tag{35}$$

Substituting the value of x_{d_i,t_j} from Eq. (34) to Eq. (35), we get

$$B_n e^{\iota(\sigma_n i \Delta d - \beta_n j \Delta t)}(e^{-\iota\beta_n 2\Delta t} - \mu e^{-\iota\beta_n \Delta t} + \chi) = 0 \tag{36}$$

Since $B_n \neq 0$ until the algorithm terminates,

$$e^{-\iota\beta_n 2\Delta t} - \mu e^{-\iota\beta_n \Delta t} + \chi = 0 \tag{37}$$

or

$$D_1^2 - \mu D_1 + \chi = 0 \tag{38}$$

where $D_1 = exp(-\iota\beta_n \Delta t)$ = amplification factor
Solving the quadratic Eq. (38), the amplification factor is obtained as

$$D_1 = \frac{\mu \pm \sqrt{\mu^2 - 4\chi}}{2}$$

Now according to von Neumann stability criterion, the finite difference scheme (32) is stable iff for the amplification factor (D_1), $|D_1| \le 1$ [41]. Therefore, we consider the finite difference scheme given by Eq. (32) and so the finite difference scheme given by Eq. (29), and hence, the PSO algorithm is stable iff $|D_1| \le 1$.

$$i.e. \quad \left| \frac{\mu \pm \sqrt{\mu^2 - 4\chi}}{2} \right| \le 1$$

$$\Rightarrow \quad -1 \leq \frac{\mu \pm \sqrt{\mu^2 - 4\chi}}{2} \leq 1,$$
$$where \quad \mu = (1 + \chi - \chi(c_1 + c_2))$$

Now the following two cases arise
Case 1:

$$\frac{\mu \pm \sqrt{\mu^2 - 4\chi}}{2} \leq 1 \tag{39}$$

or

$$\pm\sqrt{\mu^2 - 4\chi} \leq 2 - \lambda$$
$$\Rightarrow \quad |\sqrt{\mu^2 - 4\chi}| \leq 2 - \lambda \tag{40}$$

By squaring both the sides, we get

$$\mu^2 - 4\chi \leq (2 - \mu)^2 \tag{41}$$

Replacing μ by $(1 + \chi - \chi(c_1 + c_2))$ in the inequality given by Eq. (41), we get a stability condition

$$\chi(c_1 + c_2) \geq 0 \tag{42}$$

Case 2:

$$-1 \leq \frac{\mu \pm \sqrt{\mu^2 - 4\chi}}{2} \tag{43}$$

or

$$-(2 + \mu) \leq \pm\sqrt{\mu^2 - 4\chi}$$
$$\Rightarrow \quad (2 + \mu) \geq \pm\sqrt{\mu^2 - 4\chi}$$
$$\Rightarrow \quad (2 + \mu) \geq | \sqrt{\mu^2 - 4\chi} | \tag{44}$$

By squaring both the sides, we get

$$(2 + \mu)^2 \geq (\mu^2 - 4\chi) \tag{45}$$

Replacing μ by $(1 + \chi - \chi(c_1 + c_2))$ in the inequality given by Eq. (45), we get another stability condition

$$\chi(c_1 + c_2) \leq 2(1 + \chi) \tag{46}$$

From Eqs. (42) and (46), we get the condition for stability as

$$0 \leq \chi(c_1 + c_2) \leq 2(1 + \chi) \tag{47}$$

or

$$0 \leq (c_1 + c_2) \leq 2(1 + 1/\chi); \qquad \chi \neq 0 \tag{48}$$

Thus, the PSO algorithm is stable iff the acceleration coefficients and the constriction coefficient satisfy the inequality given by Eq. (48). The range of values of parameters c_1, c_2 and χ so that the inequality (47) is satisfied is termed as stable range. We denote the stable range by A_S. The compliment of this range A_S is termed as outside stable range and defined by A_{US}.

The first theorem verifies the stability condition obtained by various researchers, and second theorem proposes stability condition for PSO algorithm with constriction coefficient χ. In this study, von Neumann stability criterion for finite difference scheme is used to find stability condition of PSO algorithm with inertia weight w and constriction coefficient χ. The advantage of applying von Neumann stability criterion is that there is no need to find eigenvalues and matrix norm, so it is easy to implement. Hence, this criterion can further be used to find stability conditions of other population-based meta-heuristic search algorithms. In next section, the obtained stability condition is tested over benchmark test problems.

5 Numerical Experiments

In order to justify theoretical findings of stability analysis of PSO algorithm with constriction coefficient χ, numerical experiments are performed on ten benchmark test problems. The set of considered test problems contains uni-modal, multi-modal and separable and non-separable problems. The test problems are listed in Table 1.

To check the accuracy of PSO algorithm, numerical experiments have been carried out for cases when parameters c_1, c_2 and χ lie within stable range A_S and when they lie outside stable range A_{US_i}.

Following eight cases of parameter settings are considered while doing numerical experiment, and the results are given in Tables 2 and 3.

Case 1: When acceleration coefficients c_1, c_2 and constriction coefficient χ are in stable range A_S,

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 2)$, i.e. $c_1 \in A_S$. Here, $U(a, b)$ is a uniformly distributed random number in the interval (a, b).
6. $c_2 = U(0, 2)$, i.e. $c_2 \in A_S$.
7. χ = 0.72, i.e. $\chi \in A_S$.

Case 2: When acceleration coefficients c_1, c_2 are in unstable range A_{US} and constriction coefficient χ is in stable range A_S,

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	Characteristics
Sphere	$Minf_1(x) = \sum_{i=1}^{n} x_i^2$	$[-5.12, 5.12]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	U, S
Rastrigin	$Minf_2(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12, 5.12]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	M, S
Griewank	$Minf_3(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$	$[-600, 600]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	M, N
Alpine	$Minf_4(x) = \sum_{i=1}^{n} \lvert x_i \sin x_i + (0.1)x_i \rvert$	$[-10, 10]$	$f(\mathbf{0}) = 0$	10	$1.0E-05$	M, S
Ackley	$Minf_3(x) = -20 + e + exp(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n} x_i{}^3}) -exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi.x_i)x_i)$	$[-1, 1]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	M, S
Zakharov	$Minf_6(x) = \sum_{i=1}^{n} x_i{}^2 + (\sum_{i=1}^{n}\frac{ix_i}{2})^2 + (\sum_{i=1}^{n}\frac{ix_1}{2})^4$	$[-5.12, 5.12]$	$f(\mathbf{0}) = 0$	30	$1.0E-02$	M, N
Axis parallel hyper-ellipsoid	$Minf_7(x) = \sum_{i=1}^{n} i.x_i^2$	$[-5.12, 5.12]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	U, S
Sum of different powers	$Minf_8(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	$[-1, 1]$	$f(\mathbf{0}) = 0$	30	$1.0E-05$	U, S
Rosenbrock	$Minf_9(x) = \sum_{i=1}^{n}(100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$	$[-30, 30]$	$f(\mathbf{0}) = 0$	30	$1.0E-02$	U, N
Shifted Ackley	$Minf_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)) + 20 + e + f_{bias}$, $z = (x - o)$, $x = (x_1, x_2, \ldots x_n)$, $o = (o_1, o_2, \ldots o_n)$	$[-32, 32]$	$f(o) = f_{bias} = -140$	10	$1.0E-05$	M, N

Table 2 Mean error (ME) for region A_S and A_{US} and their comparison using Wilcoxon signed rank (WSR) test (*TP* test problem, A_S: parameters within stable range, A_{US}: parameters within unstable range)

TP	ME for Case 1	ME for Case 2 (WSR of Case 2 Vs Case 1)	ME for Case 3 (WSR of Case 3 Vs Case 1)	ME for Case 4 (WSR of Case 4 Vs Case 1)
f_1	1.94E–26	5.52E–01 (+)	7.01 (+)	3.78 (+)
f_2	37.54	101.16 (+)	190.36 (+)	185.88 (+)
f_3	1.45E–16	2.06 (+)	25.05 (+)	13.51 (+)
f_4	9.96E–14	2.63 (+)	15.36 (+)	12.45 (+)
f_5	5.45E–02	4.44 (+)	10.31 (+)	8.44 (+)
f_6	5.15E–05	2.09 (+)	11.93 (+)	6.80 (+)
f_7	5.00E–25	7.69 (+)	97.88 (+)	53.40 (+)
f_8	1.39E–41	2.98E–07 (+)	1.1E–03 (+)	2.93E–04 (+)
f_9	38.98	6.11E05 (+)	5.13E07 (+)	1.64E07 (+)
f_{10}	20.00003	20.20027 (=)	20.34508 (=)	20.31818 (=)
Number of + signs		9	9	9

Table 3 Mean error (ME) for region A_S and A_{US} and their comparison using Wilcoxon signed rank (WSR) test (**TP**: test problem, A_S: parameters within stable range, A_{US}: parameters within unstable range)

TP	ME for Case 5	ME for Case 6 (WSR of Case 6 Vs Case 5)	ME for Case 7 (WSR of Case 7 Vs Case 5)	ME for Case 8 (WSR of Case 8 Vs Case 5)
f_1	6.59E–05	1.86 (+)	7.97 (+)	3.86 (+)
f_2	31.87	170.83 (+)	210.70 (+)	168.72 (+)
f_3	7.77e–03	6.93 (+)	35.06 (+)	13.59 (+)
f_4	1.94E–01	8.22 (+)	17.38 (+)	11.46 (+)
f_5	3.81	6.78 (+)	11.59 (+)	8.51 (+)
f_6	2.24	3.38 (+)	17.35 (+)	8.27 (+)
f_7	3.55E–03	2.60E01 (+)	1.40E02 (+)	5.27E01 (+)
f_8	2.21E–16	6.80E–05 (+)	2.35E–03 (+)	3.1E–04 (+)
f_9	129.81	4.19E06 (+)	1.09E08 (+)	1.95E07 (+)
f_{10}	20	20.35 (=)	20.3382 (=)	20.3149 (=)
Number of + signs		9	9	9

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 4.50)$, i.e. $c_1 \in A_{US}$.
6. $c_2 = U(0, 3.85)$, i.e. $c_2 \in A_{US}$.
7. $\chi = 0.50$, i.e. $\chi \in A_S$.

Case 3: When acceleration coefficients c_1, c_2 are in stable range A_S and constriction coefficient χ is in unstable range A_{US},

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 1.62)$, i.e. $c_1 \in A_S$.
6. $c_2 = U(0, 1.45)$, i.e. $c_2 \in A_S$.
7. $\chi = 1.50$, i.e. $\chi \in A_{US}$.

Case 4: When acceleration coefficients c_1, c_2 are in stable range A_{US_3} and constriction coefficient χ is in unstable range A_{US},

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 4.62)$, i.e. $c_1 \in A_{US}$.
6. $c_2 = U(0, 5.45)$, i.e. $c_2 \in A_{US}$.
7. $\chi = 1.50$, i.e. $\chi \in A_{US}$.

Case 5: When acceleration coefficients c_1, c_2 and constriction coefficient χ are in stable range A_S,

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

5. $c_1 = U(0, 1.52)$, i.e. $c_1 \in A_S$.
6. $c_2 = U(0, 1.55)$, i.e. $c_2 \in A_S$.
7. $\chi = 0.50$, i.e. $\chi \in A_S$.

Case 6: When acceleration coefficients c_1, c_2 are in unstable range A_{US} and constriction coefficient χ is in stable range A_S,

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 6.52)$, i.e. $c_1 \in A_{US}$.
6. $c_2 = U(0, 5.55)$, i.e. $c_2 \in A_{US}$.
7. $\chi = 0.50$, i.e. $\chi \in A_S$.

Case 7: When acceleration coefficients c_1, c_2 are in stable range A_S and constriction coefficient χ is in unstable range A_{US},

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 1.12)$, i.e. $c_1 \in A_S$.
6. $c_2 = U(0, 1.23)$, i.e. $c_2 \in A_S$.
7. $\chi = 2.30$, i.e. $\chi \in A_{US}$.

Case 8: When acceleration coefficients c_1, c_2 are in stable range A_{US} and constriction coefficient χ is in unstable range A_{US},

1. Swarm size: 50.
2. Maximum number of runs: 100.
3. Maximum number of iterations: 1000.
4. Acceptable error: given in Table 1.
5. $c_1 = U(0, 5.62)$, i.e. $c_1 \in A_{US}$.
6. $c_2 = U(0, 4.65)$, i.e. $c_2 \in A_{US}$.
7. $\chi = 2.50$, i.e. $\chi \in A_{US}$.

Mean error (ME) is calculated for considered test problems, and numerical results are presented in Table 2 and Table 3. Numerical results are again verified by performing nonparametric test, namely Wilcoxon signed rank test, and given in Table 2 and Table 3. If the data set obtained by numerical experiments has significant difference, then it results in rejection of null hypothesis and '+' sign appears; otherwise, null hypothesis is accepted and '=' sign appears. In Table 2 and Table 3, '+' sign appears nine times out of ten. Thus, PSO algorithm performs better in terms of accuracy when parameters c_1, c_2 and χ lie within stable range A_S.

The above numerical verification of theoretical analyses explains that in order to bound generation of error in subsequent iterations, sum of acceleration coefficients must lie within the range as given in Eqs. (24) and (48).

6 Conclusion

Mathematical validation of parameter selection for stochastic algorithms has always been a challenging task. In order to bound the error generated during the iterative process of PSO algorithm, stability analysis has been carried out using von Neumann stability criterion. The condition for the stability of PSO algorithm with parameters c_1, c_2 and inertia weight (w) is obtained. It is found that the findings are same as represented in the literature using other methods. Stability condition for PSO algorithm with constriction coefficient (χ) is also obtained depending upon parameters c_1, c_2 and χ. Based on the condition, stable and unstable ranges are defined. The findings are verified by performing numerical experiments on benchmark test problems, and it is found that PSO algorithm performs better in terms of accuracy when parameters lie within stable range. Due to easy implementation of von Neumann stability criterion, it can be further used to find stability condition for various other population-based meta-heuristic algorithms.

Acknowledgements The first and second authors acknowledge the funding from South Asian University, New Delhi, India, to carry out this research. The last author (JC Bansal) gratefully acknowledges the Grant from Science and Engineering Research Board (SERB), Department of Science and Technology, Government of India, to carry out this research vide file no: MTR/2017/000820.

References

1. Abido, M.A.: Optimal power flow using particle swarm optimization. Int. J. Electr. Power Energy Syst. **24**(7), 563–571 (2002)
2. Bansal, J.C.; Gopal, A.; Nagar, A.K.: Analysing convergence, consistency, and trajectory of artificial bee colony algorithm. IEEE Access **6**, 73593–73602 (2018)
3. Bansal, J.C.; Gopal, A.; Nagar, A.K.: Stability analysis of artificial bee colony optimization algorithm. Swarm Evol. Comput. **41**, 9–19 (2018)
4. Bansal, J.C.; Sharma, H.; Jadon, S.S.; Clerc, M.: Spider monkey optimization algorithm for numerical optimization. Memet. Comput. **6**(1), 31–47 (2014)
5. Biswas, A.; Das, S.; Abraham, A.; Dasgupta, S.: Stability analysis of the reproduction operator in bacterial foraging optimization. Theor. Comput. Sci. **411**(21), 2127–2139 (2010)
6. Bonyadi, M.R.; Michalewicz, Z.: Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. IEEE Trans. Evol. Comput. **20**(3), 370–385 (2016)
7. Bonyadi, M.R.; Michalewicz, Z.: Stability analysis of the particle swarm optimization without stagnation assumption. IEEE Trans. Evol. Comput. **20**(5), 814–819 (2016)
8. Chakraborty, P.; Das, S.; Roy, G.G.; Abraham, A.: On convergence of the multi-objective particle swarm optimizers. Inf. Sci. **181**(8), 1411–1425 (2011)

9. Chuan, L.; Quanyuan, F.: The standard particle swarm optimization algorithm convergence analysis and parameter selection. In: 3rd International Conference on Natural Computation, ICNC 2007, vol. 3, pp. 823–826. IEEE (2007)
10. Clerc, M.; Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
11. Dasgupta, S.; Das, S.; Biswas, A.; Abraham, A.: On stability and convergence of the population-dynamics in differential evolution. Ai Commun. **22**(1), 1–20 (2009)
12. Esmin, A.A.A.; Lambert-Torres, G.; Alvarenga, G.B.: Hybrid evolutionary algorithm based on PSO and GA mutation. In: 6th International Conference on Hybrid Intelligent Systems, HIS'06, pp. 57–57. IEEE (2006)
13. Farivar, F.; Shoorehdeli, M.A.: Stability analysis of particle dynamics in gravitational search optimization algorithm. Inf. Sci. **337**, 25–43 (2016)
14. Fernandez-Martinez, J.L.; Garcia-Gonzalo, E.: Stochastic stability analysis of the linear continuous and discrete pso models. IEEE Trans. Evol. Comput. **15**(3), 405–423 (2011)
15. García-Gonzalo, E.; Fernández-Martínez, J.L.: Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. Appl. Math. Comput. **249**, 286–302 (2014)
16. Geem, Z.W.; Kim, J.H.; Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2), 60–68 (2001)
17. Ghodrati, A.; Lotfi, S.: A hybrid CS/PSO algorithm for global optimization. In: Asian Conference on Intelligent Information and Database Systems, pp. 89–98. Springer (2012)
18. Goldberg, D.E.: Genetic Algorithms. Pearson Education India, Delhi (2006)
19. Gopal, A.; Bansal, J.C.: Stability analysis of differential evolution. In: International Workshop on Computational Intelligence (IWCI), pp. 221–223. IEEE (2016)
20. Hegazy, O.; Soliman, O.S.; Salam, M.A.: Comparative study between FPA, BA, MCS, ABC, and pso algorithms in training and optimizing of LS-SVM for stock market prediction. Int. J. Adv. Comput. Res. **5**(18), 35 (2015)
21. Hu, X.; Eberhart, R.: Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of the 6th world multiconference on systemics, cybernetics and informatics, vol 5, pp 203–206. Citeseer (2002)
22. Jiang, M.; Luo, Y.P.; Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf. Process. Lett. **102**(1), 8–16 (2007)
23. Kadirkamanathan, V.; Selvarajah, K.; Fleming, P.J.: Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans. Evol. Comput. **10**(3), 245–255 (2006)
24. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
25. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of machine learning, pp. 760–766. Springer (2011)
26. Liu, B.; Wang, L.; Jin, Y.-H.; Tang, F.; Huang, D.-X.: Improved particle swarm optimization combined with chaos. Chaos Solit Fract **25**(5), 1261–1271 (2005)
27. Liu, Q.: Order-2 stability analysis of particle swarm optimization. Evol. Comput. **23**(2), 187–216 (2015)
28. Mendes, R.; Cortez, P.; Rocha, M.; Neves, J.: Particle swarms for feedforward neural network training. In: Proceedings of the 2002 international joint conference on neural networks, IJCNN'02, vol 2, pp. 1895–1899. IEEE (2002)
29. Premalatha, K.; Natarajan, A.M.: Hybrid PSO and GA for global maximization. Int. J. Open Problems Compt. Math **2**(4), 597–608 (2009)
30. Rapaić, M.R.; Kanović, Ž.: Time-varying pso–convergence analysis, convergence-related parameterization and new parameter adjustment schemes. Inf. Process. Lett. **109**(11), 548–552 (2009)
31. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
32. Reyes-Sierra, M.; Coello, C.A.C.; et al.: Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int. J. Comput. Intell. Res. **2**(3), 287–308 (2006)
33. Richtmyer, R.D.; Morton, K.W.: Different methods for initial value problems. Interscience tracts in pure and applied mathematics, no. 4, vol. 10. Interscience Publishers, New York (1967)
34. Salman, A.; Ahmad, I.; Al-Madani, S.: Particle swarm optimization for task assignment problem. Microprocess. Microsyst. **26**(8), 363–371 (2002)
35. Shi, Y.; Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on/IEEE World Congress on Computational Intelligence on Evolutionary Computation Proceedings, pp 69–73. IEEE (1998)
36. Storn, R.; Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
37. Sun, J.; Xiaojun, W.; Palade, V.; Fang, W.; Lai, C.-H.; Wenbo, X.: Convergence analysis and improvements of quantum-behaved particle swarm optimization. Inf. Sci. **193**, 81–103 (2012)
38. Thomas, J.W.: Numerical partial differential equations: finite difference methods, vol. 22. Springer, Berlin (2013)
39. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Process. Lett. **85**(6), 317–325 (2003)
40. Van der Merwe, D.W.; Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: The 2003 Congress on Evolutionary Computation, CEC'03, vol. 1, pp. 215–220. IEEE (2003)
41. Warming, R.F.; Hyett, B.J.: The modified equation approach to the stability and accuracy analysis of finite-difference methods. J. Comput. Phys. **14**(2), 159–179 (1974)
42. Zhang, C.; Shao, H.; Li, Y.: Particle swarm optimisation for evolving artificial neural network. In: IEEE International Conference on Systems, Man, and Cybernetics, 2000, vol. 4, pp. 2487–2490. IEEE (2000)