frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Methods
PUBLISHED 14 February 2024
DOI 10.3389/fams.2024.1304268

OPEN ACCESS

EDITED BY
Guillermo Huerta Cuellar,
University of Guadalajara, Mexico

REVIEWED BY
Hector Eduardo Gilardi Velazquez,
Universidad Panamericana, Mexico
Felipe Medina-Aguayo,
Instituto Tecnológico Autónomo de México, Mexico

*CORRESPONDENCE
Dereje Tarekegn Nigatu
✉ tekgem@yahoo.com

RECEIVED 29 September 2023
ACCEPTED 18 January 2024
PUBLISHED 14 February 2024

CITATION
Tarekegn Nigatu D, Gemechu Dinka T and Luleseged Tilahun S (2024) Convergence analysis of particle swarm optimization algorithms for different constriction factors. *Front. Appl. Math. Stat.* 10:1304268. doi: 10.3389/fams.2024.1304268

Convergence analysis of particle swarm optimization algorithms for different constriction factors

Dereje Tarekegn Nigatu[1]*, Tekle Gemechu Dinka[1] and Surafel Luleseged Tilahun[2]

[1]Department of Applied Mathematics, Adama Science and Technology University, Adama, Ethiopia, [2]Department of Applied Mathematics, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia

Particle swarm optimization (PSO) algorithm is an optimization technique with remarkable performance for problem solving. The convergence analysis of the method is still in research. This article proposes a mechanism for controlling the velocity by applying a method involving constriction factor in standard swarm optimization algorithm, that is called CSPSO. In addition, the mathematical CSPSO model with the time step attractor is presented to study the convergence condition and the corresponding stability. As a result, constriction standard particle swarm optimization that we consider has a higher potential to balance exploration and exploitation. To avoid the PSO premature convergence, CSPSO modifies all terms of the PSO velocity equation. We test the effectiveness of the CSPSO algorithm based on constriction coefficient with some benchmark functions and compare it with other basic PSO variant algorithms. The theoretical convergence and experimental analyses results are also demonstrated in tables and graphically.

KEYWORDS

PSO algorithms, convergence and stability, constriction factor, Markov chain, Monte Carlo

1 Introduction

The optimization techniques are fundamentally important in engineering and scientific computing. The PSO algorithm was first introduced by Kennedy and Eberhart [1] as a stochastic optimization technique of swarm particles (population). The motivation was primarily to model the social behavior of birds flocking. The meta-heuristic optimization algorithms (PSO) work effectively in many areas such as robotics, wireless networks, power systems, job-shop schedules, human healthcare, and classifying or training of ANN (artificial neural network) [2]. In PSO, the potential solutions, called particles, fly through the problem space (domain) by applying their intelligent collective behaviors.

The PSO algorithm is competitive in performance with the well-known huge numbers of variants such as SPSO and CPSO algorithms and is also an efficient optimization framework [3, 4].

Lately, researches on PSO mainly intended on algorithmic implementations, enhancements, and engineering applications with interesting findings derived under the system that assumes a fixed attractor [5]. Nevertheless, a comprehensive mathematical explanation for the general PSO is still quite limited. For instance, the works on stability and convergence analyses are two key problems of great significance that need to be investigated in depth because many of the works have given attention for standard PSO.

The PSO algorithm depends on three parameters (factors): the inertia, cognitive and social weight to guarantee the stability of PSO.

Stability analysis of PSO is mainly motivated by determining which combination of these parameters encourages convergence [6].

The working rule of PSO method is closely tied with the stability analysis, which investigates how the essential factors affect the swarms dynamics, and under what conditions particle swarm converges to some fixed value. For the first time, stability analysis of the particle dynamics was carried out by Clerc and Kennedy [7]. The study indicates that [8] particle trajectories could converge to a stable point. A more generalized stability analysis of the particle dynamics was conducted using the Lyapunov stability theorem [9]. Recently, based on a weak stagnation assumption, Liu [10] studied the order-2 stability of PSO, and a new definition of stability was proposed with an order-2 stable region. Dong and Zhang [11] analyzed order-3 recurrence relation of PSO kinematic equations based on two strategies to obtain the necessary and sufficient conditions of its convergence.

The convergence analysis determines whether a global optimum solution can be achieved when a particle swarm converges. Using stochastic process theory, Jiang et al. [12] presented a stochastic convergence analysis on the standard PSO. Combining with the finite element grid technique, Poli and Langdon [13] set up a discrete model of Markov chain of the bare-bones PSO. An absorbing Markov process model of PSO was developed in Cai et al. [14]. Cai et al. [14] suggested the main factor of convergence analysis is the attaining-state set and proposed an improved method of convergence in terms of the attaining-state set theorem of expansion. The basic PSO is neither a global nor a local search algorithm, based on the convergence criterion of the pure random search algorithm [15, 16]. To yield a lower bound for the time required to optimize any pseudo-Boolean functions with a unique optimum and to justify upper bounds, Dirk et al. [17] assigned an optimum-level argument that is deep-rooted for evolutionary algorithms of particle swarm optimization. The study in Sun et al. [18] discussed the convergence of the quantum-behaved particle swarm optimization (QBPSO) and proved that it is a global convergent algorithm.

As discussed in Per and Carsten [19], stagnation of the convergence properties for basic PSO may be disadvantageous to finding a sufficiently good solution within a logical time, and it may have infinite expected first hitting time on some functions.

Recently, the existing work on the convergence analyses of PSO including documents from 2013 was surveyed by Tarekegn et al. [6]. The stochastic approximation technique on the PSO algorithm was use to prove convergence of swarm in Yuan and Yin [20]. The global convergence of PSO [21] was investigated by introducing the transition probability of particles. Several properties related to the Markov chain were investigated, and it was found that the particle state space is not repeated and PSO is not globally convergent from the viewpoint of the transition probability [22]. Based on the different models of PSO examined [23], the Markov properties of the state sequences of a single particle and swarm one determine the transition probability of a particle. The transition probability of the optimal set is deduced by combining the law of total probability with the Markov properties [24], which proves that SPSO can reach the global optimum in probability. Although many methods in Poli and Langdon [13] have proposed PSO convergence analysis, most analyses are based on the assignment of stochastic systems of the Markov process, which strongly depends on the transition matrix and their eigenvalues. Therefore, when the population size is large, current PSO convergence analyses are very refined and investigate different PSO variants algorithms to obtain a solution that converges to global minimum.

Motivated by our recent study in Tarekegn et al. [6], this article proposes a PSO variant known as CSPSO, an algorithm for optimization problem solving.

A constriction factor integrated with an inertia weight are used for the construction. Fast convergent method to an optimal solution within the search space in a small time of iterations was obtained.

The rest of this study is organized as follows: Section 2 presents related works that include the basic PSO algorithm and its existing variants. In Section 3, the proposed CSPSO algorithm analysis is described in detail, while Section 4 presents comparison results on some variants of PSO such as SPSO and CPSO (implementing with test functions) and provides an in-depth discussion, with a conclusion in Section 5.

2 The PSO algorithm and some related studies

In the PSO with K particles in which each particle is treated as an individual in the D-dimensional space, the position and velocity vectors of the i-th particle at the t-th iteration are

$X_i^t = (X_{i1}^t, X_{i2}^t, \ldots, X_{iD}^t)$ and

$V_i^t = (V_{i1}^t, V_{i2}^t, \ldots, V_{iD}^t)$, respectively.

In SPSO algorithm [25], at iteration t, the d th dimension of particle i's velocity and position P_i^t is local best position, x_i^t is current position, and g^t is global best position. Both are updated as

$$\begin{aligned} V_i^{t+1} &= \omega V_i^t + c_1 r_1^t (P_i^t - X_i^t) + c_2 r_2^t (g^t - X_i^t), \\ X_i^{t+1} &= X_i^t + V_i^{t+1}, \end{aligned} \tag{1}$$

for $1 \leq i \leq K$; ω is an inertia weight; and c_1 and c_2 are called acceleration coefficients in real-space, R.

Vector $P_i^t = (P_{i1}^t, P_{i2}^t, \ldots, P_{iD}^t)$ is the best previous position of particle i called personal best (Pbest) position and vector $g^t = (g_1^t, g_2^t, \ldots, g_D^t)$ is the position of the best particle among all the particles in the population and called global best (gbest) position. The parameters r_1^t and r_2^t are sequences of two different random positive numbers in the uniform random distribution in (0, 1) i.e., $U(0, 1)$.

Generally, the value of V_{id}^t is restricted within the interval $[-V_{max}, V_{max}]$, for each $d \in \{1, 2, \ldots, D\}$. Without loss of generality, we consider the minimization problem:

Minimize $f(X)$, such that

$$X \in S \subset \mathbb{R}^D, \tag{2}$$

where $f(X)$ is an objective function continuous almost everywhere and S is a feasible solution space. From (1),

the non-homogeneous recurrence relation (NHRR) is obtained as follows: [8]

$$X_i^{t+1} = -\omega x_i^{t-1} + (1+\omega)x_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t), \tag{3}$$

where $\varphi_1^t = c_1 r_1^t$, $\varphi_2^t = c_2 r_2^t$.

From NHRR, P^t and g^t, for $1 \leq i \leq K$, are updated, respectively, as follows:

$$P_i^{t+1} = \begin{cases} x_i^{t+1} : for\, f(x_i^{t+1}) < f(P_i^t) \\ P_i^t \quad : otherwise; \end{cases} \tag{4}$$

$$g^0 = \underset{1\leq i\leq k}{arg\ min}\{f(x_i^o)\},$$
$$g^{t+1} = \underset{1\leq i\leq k}{arg\ min}\{f(x_i^{t+1}), f(g^t)\}.$$

From (2), (3), the process of the particle's velocity and position change can be obtained, respectively, as follows. They are a second-order difference equations

$$V_i^{t+2} + (\varphi^t - \omega - 1)V_i^{t+1} + \omega V_i^t = 0 \tag{5}$$

$$\begin{aligned} X_i^{t+1} + (\varphi^t - \omega - 1)x_i^t + \omega x_i^{t-1} &= \varphi_1^t P_i^t + \varphi_2^t g^t \\ &= \varphi^t O_i^t \end{aligned} \tag{6}$$

where, $\varphi^t = \varphi_1^t + \varphi_2^t$

$$O_i^t = \frac{\varphi_1^t p_i^t + \varphi_2^t g^t}{\varphi^t}. \tag{7}$$

The terms $(\varphi^t - \omega - 1)x_i^t$ and ωx_i^{t-1} on the left side of (6), both memorize the past values of position (i.e, the memory item of position). The value of the item $\varphi^t O_i^t$ on the right side of (6) is obtained from the previous experience of particles (i.e, the learning item of position) and, in particular, O_i^t is the attractor at the t th iteration in (7).

Now, let

$$Q_x = \underset{x^t \in S_x \subset R}{max} \mid x(t) \mid . \tag{8}$$

For p^t, $g^t \in S_x$, $|p^t| \leq Q_x$, and $|g^t| \leq Q_x$. From (8), $O^t \in S_o$ means $|O^t| \leq Q_o$ for all t.

Introducing a constriction coefficient in SPSO controls the balance between the cognitive component $(p_i^t - x_i^t)$ and social component $(g^t - x_i^t)$ in the velocity equation. The coefficient restricts the particle velocities within a certain range to prevent excessive exploration or exploitation.

$$V_i^{t+1} = \chi * \left(\omega V_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)\right)$$

2.1 Convergence of some PSO variants

The importance of a hybrid method is to combine different optimization methods to take advantage of the virtues of each of the methods. In addition to standard PSO, several variants of the PSO in Kumar et al. [5] were constructed to improve the performance of PSO.

The SPSO

$$X_i^{t+1} = X_i^t + V_i^{t+1},$$

has a scalar function of position if $x_i^t = p_i^t = g_i^t$ for a particle, that is particle's update depends only on its previous velocity. This can make the algorithm to stop to flow on the swarm's global best position, even if that position is not a local optimum. For instance, based on (4), the guaranteed convergence PSO, GCPSO, overcomes this problem by using a modified position and velocity update equation for the global best particle, which forces that particle to search for a better position in a confined region around the global best position.

The GCPSO can be used with neighborhood topologies such as star, ring, and Von Neumann. Neighborhoods have a similar effect in the GCPSO [16, 19] as they do in the SPSO. Shi and Eberhart [25] introduced the concept of linearly decreasing inertia weight with generation number into PSO to improve the algorithmic performance.

Particles converge to a weighted average (O_i^t) between their personal and local best positions [8], referred to as a so-called theoretical attractor point (ATP). Kennedy [26] has proposed that the entire velocity update equation is replaced by a random number sampled from a Gaussian distribution (Gd) around the ATP, with a deviation of the magnitude of the distance between the personal and global best. The resultant algorithm is called the bare bones PSO (BBPSO). Kennedy also proposed an alternative bare bones PSO (aBBPSO) [26], where the particle sampled from the previous Gd is reunited with the particle's personal best position. The performance of PSO with a small and a larger nearby region might be better on multimodal and unimodal problems, respectively [27]. Changing dynamically the neighborhood structures has been proposed to avoid insufficiencies in fixed nearby regions [28].

The quantum-behaved particle swarm optimization was proposed to show many advantages to the traditional PSO. Fang et al. [24] proposed a quantum-behaved particle swarm optimization (QBPSO) algorithm and discuss the convergence of QBPSO within the framework of random algorithm's global convergence theorem. Inspired by natural speciation, some researchers have introduced evolution methods into PSO [29, 30]. The problem of premature convergence was studied on a perturbed particle swarm algorithm presented based on the new particle updating strategy [31]. To solve optimization problems, Tang et al. [32] developed a feedback-learning PSO algorithm with quadratic inertia weight, ω. Hybridized PSO with a local search technique for locating optimal solutions for multiple global and local solution in physical fitness of more than one global optimal solution for optimization problem using a memetic algorithm can be referred in Wang et al. [33]. An example-based learning PSO was proposed in Huang et al. [34] to overcome the failures of PSO by retaining a balance between swarm diversity and convergence speed. A

variation of the global best PSO where the velocity update equation does not hold a cognitive component is called social PSO, expressed as

$$V_i^{t+1} = \omega V_i^t + \varphi_2(g_i^t - x_i^t), \tag{9}$$

The individuals are only supported by the global best position and their previous velocity. The particles are attracted toward the global best position, instead of a weighted average between global best and their personal best positions, leading to very fast convergence [19].

3 Relations of CSPSO and Markov chain

In this section, the global convergence of CSPSO is analyzed based on properties of Markov Chain and the transition probabilities of particle velocity and position are also computed.

$$V_i^{t+1} = \begin{cases} \chi * (V_i^t + \varphi_1^t(P_i^t - x_i^t) + \\ \qquad \varphi_2^t(g^t - x_i^t)), \ for\ \omega = 1 \\ \chi * (\omega V_i^t + \varphi_1^t(P_i^t - x_i^t) + \\ \qquad \varphi_2^t(g^t - x_i^t)), \ otherwise \end{cases} \tag{10}$$

In (10), the velocities of particles are updated using two main components: the cognitive component and the social component. The cognitive component guides a particle toward its personal-best position, while the social component directs a particle toward the best position found by the entire swarm.

We introduce some useful definitions, variables and propositions (based on single particle model) which may be important in this article [22, 23, 35, 36].

The following definitions provide a formal description of this property based on single particle model [22, 23, 35, 36].

Definition 1. (Stochastic process and Markov property). Assume all the variables are defined within the context of a common probability space or probability measure.

1. The random variables $Y = (Y^0, Y^1, \ldots, Y^t)$ in a sequence are called a stochastic processes.
2. Let Y^t be a value in state space S, and the sequence $\{Y^t\}_{t\geq 0}$ is a discrete stochastic process.
   For every $t \geq 0$ and $i^l \in S(l-1 \leq t)$.
3. The discrete stochastic process is a Markov Chain.
   If the probability $Pr\{Y^{t+1} = i^{t+1} \mid Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} = Pr\{Y^{t+1} = i^{t+1} \mid Y^t = i^t\} > 0$. and $Pr\{Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} > 0$

Definition 2. (State of particle). The state of particle $\kappa_i^t = (x_i^{t-1},\ x_i^t,\ p_i^t,\ g^t)$ at the t-th iteration for particle i in (3).

The state of particle space is a set of all possible states of particle, denoted as S. κ_i^t, the update probability of the state of the particle can be calculated based on proposition-1.

Proposition 1. If the accelerating factors φ_1^t and φ_2^t in CSPSO satisfy $\varphi_1^t,\ \varphi_2^t \in U(0, c)$, then the probability for particle i changes from the position x_i^t to the spherical region centered at x_i^{t+1} with radius $\varrho_t > 0$. The event $A_i = \{\kappa_i^{t+1} \mid \kappa_i^t\}$, defining the state of particle i at the t-th iteration is updated to the state at the $(t+1)$-th iteration, for each $i \in \{1, 2, \ldots, K\}$ can be computed as

$$Pr(A_i) = \frac{\varrho_t^3}{\chi\omega \mid\mid x_i^t - X_i^{t-1} \mid\mid\ c\chi \mid\mid p_i^t - X_i^t \mid\mid c\chi \mid\mid g^t - X_i^t \mid\mid}, \tag{11}$$

c is a constant within $U(0, c)$ and $\delta \to 0$, where

$$\varrho = c\chi * \begin{cases} \mid\mid p_i^t - X_i^t \mid\mid, for\, f(x_i^t) - \delta \leq f(P_i^t) \leq f(x_i^t) + \delta \\ \mid\mid g^t - X_i^t \mid\mid, for\, f(x_i^t) - \delta \leq f(g^t) \leq f(x_i^t) + \delta \end{cases} \tag{12}$$

Proof. The 1-step transition probability of the i th state of particles, P_i^{t+1} and g^{t+1}, are determined by x_i^{t+1} for transferring κ_i^t to κ_i^{t+t} based on the following SPM-Single Particle Model [36]

$$\begin{aligned} x_i^{t+1} = x_i^t + \chi * (\omega(x^t - x_i^{t-1}) + \\ \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)), \end{aligned} \tag{13}$$

x_i^{t+1} determined by $\chi\omega,\ \chi\varphi_1$, and $\chi\varphi_2$.

Three conditions in 1-step transition probability are:

1. $f(x_i^t) - \delta \leq f(P_i^t),\ f(g^t) \leq f(x_i^t) + \delta$. $x_i^{t+1} = x_i^t + \chi\omega(x_i^t - x_i^{t-1})$ is determined uniquely by $\chi\omega$, where $\chi\omega$ is unknown constant, having

$$\begin{aligned} P(x_i^{t+1} \mid \kappa_i^t) &= \frac{\int_{x_i^{t+1} - \frac{1}{2}\varrho}^{x_i^{t+1} + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t - \chi\omega(x_i^t - x_i^{t-1})} dy} \\ &= \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid} \end{aligned} \tag{14}$$

2. $f(x_i^t) - \delta \leq f(P_i^t),\ f(g^t) \leq f(x_i^t) + \delta$.
   Ordering implies $g^t \in \{p_i^t,\ g^t\}$

$$\begin{aligned} x_i^{t+1} = x_i^t + \chi\omega(x^t - x_i^{t-1}) + \\ \chi\varphi_2^t(g^t - x_i^t) \end{aligned} \tag{15}$$

Here, φ_2^t is random variable because x_i^{t+1} is determined by $\chi\omega(x^t - x_i^{t-1})$ and $\chi\varphi_2^t(g^t - x_i^t)$

$$\begin{aligned} P(x_i^{t+1} \mid \kappa_i^t) &= \frac{\int_{x_i^{t+1} - \frac{1}{2}\varrho}^{x_i^{t+1} + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t + \chi\omega(x_i^t - x_i^{t-1})} dy} * \frac{\int_{\varphi_1^t - \frac{1}{2}\varrho}^{\varphi_1^t + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t + \chi c(g^t - x_i^{t-1})} dy} \\ &= \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^t \mid\mid} * \frac{\varrho}{\chi c \mid\mid g^t - x_i^t \mid\mid} \end{aligned} \tag{16}$$

3. $f(x_i^t) + \delta < f(P_i^t),\ f(g^t) < f(x_i^t) - \delta$.

$$\begin{aligned} x_i^{t+1} = x_i^t + \chi\omega(x^t - x_i^{t-1}) + \chi\varphi_1^t(P_i^t - x_i^t) \\ + \chi\varphi_2^t(g^t - x_i^t)) \end{aligned} \tag{17}$$

when $x_i^{t+1} \in \mathbb{R}$, x_i^{t+1} is determined by $\chi * (\omega, \varphi_1, \varphi_2)$

$$P(x_i^{t+1} \mid \kappa_i^t) = \frac{\int_{x_i^{t+1}-\frac{1}{2}\varrho}^{x_i^{t+1}+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi\omega(x_i^t-x_i^{t-1})} dy} * \frac{\int_{\varphi_1^t-\frac{1}{2}\varrho}^{\varphi_1^t+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi c(g^t-x_i^{t-1})} dy} * \frac{\int_{\varphi_2^t-\frac{1}{2}\varrho}^{\varphi_2^t+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi c(g^t-x_i^{t-1})} dy} = \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid} * \frac{\varrho}{\chi c \mid\mid P_i^t - x_i^t \mid\mid} * \frac{\varrho}{\chi c \mid\mid g^t - x_i^t \mid\mid} \tag{18}$$

From conditions in $1-3$,

$$\lim_{P_i^t \to x_i^t} \frac{\varrho}{\chi\omega \mid\mid P_i^t - x_i^t \mid\mid} = 1$$

$$\lim_{g^t \to X_i^t} \frac{\varrho}{\chi c \mid\mid g_i^t - x_i^t \mid\mid} = 1.$$

$$P(A_i) = \frac{\varrho^3}{\chi(\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid c \mid\mid P_i^t - x_i^t \mid\mid c \mid\mid g^t - x_i^t \mid\mid} \tag{19}$$

δ is a vector approaching to zero. When

I. ϱ is the radius of x_i^{t+1}
II. $f(x_i^t) - \delta \le f(P_i^t) \le f(x_i^t) + \delta$, $\varrho = c\chi \mid\mid P_i^t - x_i^t \mid\mid$
III. $f(x_i^t) - \delta \le f(g^t) \le f(x_i^t) + \delta$, $\varrho = c\chi \mid\mid g^t - x_i^t \mid\mid$

Definition 3. (State of swarm). The state of swarm in (3), at iteration t, denoted as η^t, is defined as $\eta^t = (\kappa_1^t, \kappa_2^t, \ldots, \kappa_K^t)$.
The state of swarm space is a set of all possible states of swarm, denoted as ϖ [22].

Proposition 2. (Markov chain). The set of collection of swarm state $\{\eta^t\}_{t\ge1}$ is a Markov chain[23].

Proof. The proof follows by referring to equation of position that the state of swarm $\eta^{t+1} = (\kappa_1^{t+1}, \kappa_2^{t+1}, \ldots, \kappa_m^{t+1})$ at iteration $t+1$ depends on only the state of swarm $\eta^t = (\kappa_1^t, \kappa_2^t, \ldots, \kappa_K^t)$. at iteration t. Therefore, $\{\eta^t\}_{t\ge1}$ is a Markov chain.

Definition 4. Let Γ_1^n denote the σ-field generated by particles state $\kappa_1^t, \kappa_2^t, \ldots, \kappa_n^t$, $(K \ge n)$ and define

$$\phi((\Gamma_1^n, \kappa_{n+1}^t) = sup\{\mid Pr(B\backslash A) - Pr(B) \mid : A \in \Gamma_1^n, B \in \sigma(\kappa_{n+1}^t)\}, \tag{20}$$

$$\phi = \sup_{1\le n\le K-1} \phi(\Gamma_1^n, \kappa_{n+1}^t) \tag{21}$$

Due to the weak interdependent relationship among the particles, ϕ is approximately small.

Proposition 3. The transition probability from η^t to η^{t+1} satisfies

$$\mid Pr(\eta^{t+1} \mid \eta^t) - \prod_{i=1}^{K} Pr(\kappa_i^{t+1} \mid \kappa_i^t) \mid \le \mu \tag{22}$$

where μ can be made small enough, therefore, $\mu = (2^{K-1}-1)\phi$.

Proof. Based on the Definition 4, one has $\mid Pr(B\backslash A) - Pr(B) \mid \le \phi$.

The event $\{\kappa_i^{t+1} \mid \kappa_i^t\}$ denoted as A_i means that the state of particle i at the t-th iteration is changed to the state at the $(t+1)$-th iteration, for each $i \in \{1, 2, \ldots, K\}$.

$$Pr(\prod_{i=1}^{K} A_i) = Pr(\eta^{t+1} \mid \eta^t) \tag{23}$$

is the transition probability from η^t to η^{t+1}.

Because g^{t+1} depends on x_i^t and P_i^t for all $1 \le i \le K$, $A_1, A_2, \ldots, A_K$ are not independent random events.
According to (6) and the conditional probability, one has the following cases:

Case 1: $Pr(A_1A_2) = Pr(A_1)Pr(A_2 \mid A_1) \le Pr(A_1)[P(A_2) + \phi] \le Pr(A_1)Pr(A_2) + \phi,$

Case 2: $Pr(A_1)Pr(A_2) - \phi \le Pr(A_1)[Pr(A_2) - \phi] \le Pr(A_1)Pr(A_2 \mid A_1) = Pr(A_1A_2).$

This implies

$$Pr(A_1)Pr(A_2) - \phi \le Pr(A_1A_2) \le Pr(A_1)Pr(A_2) + \phi \tag{24}$$

Case 3: $Pr(A_1A_2A_3) = Pr(A_1)Pr(A_2 \mid A_1)Pr(A_3 \mid A_1A_2) \le Pr(A_1)[Pr(A_2) + \phi][Pr(A_3) + \phi] \le Pr(A_1)Pr(A_2)Pr(A_3) + 3\phi,$

Case 4: $Pr(A_1)Pr(A_2)Pr(A_3) - 3\phi \le Pr(A_1)[Pr(A_2) - \phi][Pr(A_3) - \phi] \le Pr(A_1)Pr(A_2 \mid A_1)Pr(A_3 \mid A_1A_2) = Pr(A_1A_2A_3)$

Similarly, we can get

$$Pr(A_1)Pr(A_2)P(A_3) - 3\phi \le Pr(A_1A_2A_3) \le Pr(A_1)Pr(A_2)Pr(A_3) + 3\phi \tag{25}$$

Then,

$$\prod_{i=1}^{K} Pr(A_i) - (2^{K-1}-1)\phi \le Pr(\prod_{i=1}^{K} A_i) \le \prod_{i=1}^{K} Pr(A_i) - (2^{K-1}-1)\phi \tag{26}$$

is the transition probability from η^t to η^{t+1}. Let $\mu = (2^{K-1}-1)\phi$. We have

$$\mid Pr(\eta^{t+1} \mid \eta^t) - \prod_{i=1}^{K} P(\kappa_i^{t+1} \mid \kappa_i^t) \mid \leq \mu \tag{27}$$

From (1), the interdependent relationship among the particles is weak, ϕ in (11) is sufficiently small so that the fact that K is finite implies that μ is a small enough positive number.

3.1 Probabilistic convergence analysis of CSPSO

In this subsection, we present the convergence analysis for the version of the standard PSO with constriction coefficient (CSPSO), by analogy of the method of analyzing convergence of the PSO convergence of the PSO in Kennedy and Mendes [27]. We also based on concepts of definitions and results in Section 3 above. Our analysis has the advantage of providing a much easier method to realize the convergence of the PSO with constriction coefficient (χ) in comparison to the original analysis [12]. To conduct the convergence analysis of the SPSO with constriction coefficient (CSPSO), we consider the time step value $\triangle\ \tau$ to describe the dynamics of the PSO, and rewrite the velocity and position update formulas in (1) as follows:

$$\begin{aligned} V_i^{t+1} =& \chi * \{\omega V_i^t + \varphi_1 \frac{(P_i^t - x_i^t)}{\triangle\ \tau} + \\ & \varphi_2 \frac{(g_i^t - x_i^t)}{\triangle\ \tau}\}, \end{aligned} \tag{28}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \triangle\ \tau, \tag{29}$$

By replacing (28) into (29), we obtain the following probabilistic CSPSO:

$$\begin{aligned} X_i^1 = X_i + \chi * \{\omega V_i + \varphi_1 \frac{(P_i - x_i)}{\triangle\ \tau} + \\ \varphi_2 \frac{(g_i - x_i)}{\triangle\ \tau}\} \triangle\ \tau, \end{aligned} \tag{30}$$

$$\begin{aligned} X_i^1 = X_i + \chi\omega V_i \triangle\ \tau + \\ \chi\varphi \left(\frac{\chi\varphi_1 P_i + \chi\varphi_2 g_i}{\chi\varphi} - x_i \right) \end{aligned} \tag{31}$$

By rearranging the terms in (31), we obtain

$$\begin{aligned} X_i^1 = (1 - \chi\varphi) X_i + \chi\omega \triangle\ \tau V_i + \\ \chi\varphi_1 P_i + \chi\varphi_2 g_i. \end{aligned} \tag{32}$$

In addition, by rearranging the terms in (29), we obtain

$$\begin{aligned} V_i^1 = -\frac{\chi\varphi}{\triangle\ \tau} X_i + \chi\omega V_i + \\ \chi\varphi_1 \frac{P_i}{\triangle\ \tau} + \chi\varphi_2 \frac{g_i}{\triangle\ \tau}. \end{aligned} \tag{33}$$

We combine the above two (32), (33) to have the following matrix form:

$$\begin{aligned} \begin{pmatrix} X_i^1 \\ V_i^1 \end{pmatrix} = \begin{pmatrix} 1 - \chi\varphi & \chi\omega\triangle\tau \\ -\frac{\chi\varphi}{\triangle\tau} & \chi\omega \end{pmatrix} \begin{pmatrix} X_i \\ V_i \end{pmatrix} + \\ \begin{pmatrix} \chi\varphi_1 & \chi\varphi_2 \\ \frac{\chi\varphi_1}{\triangle\tau} & \frac{\chi\varphi_2}{\triangle\tau} \end{pmatrix} \begin{pmatrix} P_i \\ g_i \end{pmatrix} \end{aligned} \tag{34}$$

which can be thought of as a discrete dynamic system representation for the PSO in which $(X \quad V)^T$ is the state subject to an external input $(P_i \quad g_i)^T$, and the two terms on the right side of the equation correspond to the dynamic and input matrices, respectively [37].

Supposing that no external excitation exists in the dynamic system, $[P_i, g_i]^T$ is constant, i.e., other particles cannot find better positions. Then, a convergent behavior could be maintained. If it converges as $\tau \to \infty$, $(X_i^1 \quad V_i^1)^T \to (X_i \quad V_i)^T$. That is, the dynamic system becomes:

$$\begin{aligned} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 - \chi\varphi & \chi\omega\triangle\tau \\ -\frac{\chi\varphi}{\triangle\tau} & \chi\omega \end{pmatrix} \begin{pmatrix} X_i \\ V_i \end{pmatrix} + \\ \begin{pmatrix} \chi\varphi_1 & \chi\varphi_2 \\ \frac{\chi\varphi_1}{\triangle\tau} & \frac{\chi\varphi_2}{\triangle\tau} \end{pmatrix} \begin{pmatrix} P_i \\ g_i \end{pmatrix} \end{aligned}$$

which holds only when $V_i = 0$ and $X_i = P_i = g_i$, where the convergent point is an equilibrium point if there is no external excitation, but better points are found by the optimization process with external excitation. For (34), Tarekegn et al. [6] has mentioned a sufficient strategies of improved convergence via theoretical analysis to get the relationship among χ, ω, and φ at the condition of convergence.

The derived probabilistic CSPSO can utilize any probabilistic form of prior information in the optimization process and, therefore, the benefits from prior information can lead probabilistic CSPSO to more probable search region and help optimize more quickly with hierarchical use of parameters [40].

By substituting (28) into (29), is transformed into (35)

$$V_i^1 = \chi\omega V_i + \frac{\chi\varphi}{\triangle\tau} \left[P(i) - x_i \right] \tag{35}$$

where $P(i) = \frac{\varphi_1 P_i + \varphi_2 g_i}{\varphi}$.

Let $y_i = P(i) - x_i$, then (32), (35) can be transformed into (37), (38)

$$V_i^1 = \chi\omega V_i + \frac{\chi\varphi}{\triangle\tau} y_i \tag{36}$$

$$Y_i^1 = \chi\omega V_i + \left(1 + \frac{\chi\varphi}{\triangle\tau} \right) y_i \tag{37}$$

Combining (36) an iterative equation in the form of vector is obtained as (38)

$$\begin{pmatrix} V_i^1 \\ Y_i^1 \end{pmatrix} = \begin{pmatrix} \chi\omega & \frac{\chi\varphi}{\triangle\tau} \\ \chi\omega & 1 + \frac{\chi\varphi}{\triangle\tau} \end{pmatrix} \begin{pmatrix} V_i \\ y_i \end{pmatrix} \tag{38}$$

which can be viewed as a general forecasting model of Markov chain as follows: $\eta^t = PK_i^t$, where, K_i^t is a vector as presented below

$$(\eta^t)^T = \begin{bmatrix} V_i^1 & Y_i^1 \end{bmatrix}$$

$$P^T = \begin{bmatrix} \chi\omega & \chi\omega \\ \frac{\chi\varphi}{\triangle\tau} & 1 + \frac{\chi\varphi}{\triangle\tau} \end{bmatrix},$$

$$(K_i^t)^T = \begin{bmatrix} V_i & y_i \end{bmatrix}$$

(38) is the model with no external excitation, which is useful in studying the evolution of certain systems over repeated trials as a probabilistic (stochastic) model [37].

Using the Markov chain method, the position $(\eta^t)^T(t+1)$ of the d th element of the i th particle at the $(t+1)$ th iteration in CSPSO can be computed using the following formula:

$$(\eta^t)^T(t+1) = [(k_i^t)^T(t)]XP^T \tag{39}$$

superscript T denotes the transposition.

Based on [20, 41] the CSPSO algorithm analysis in Markov chain theory, the algorithm satisfies the context of almost sure convergence as follows:

1. As the algorithm progresses and more iterations are performed, it will converge to an optimal solution with a probability of 1 and
2. Given sufficient time and iterations, it will find the globally optimal solution.

3.2 Stability analysis of CSPSO

We further get insight into the dynamic system in (39). First, we solve the characteristic equation of the dynamic system as follows:

$$\lambda^2 - (1 + \chi\omega + \frac{\chi\varphi}{\triangle\tau})\lambda + \chi\omega = 0.$$

The eigenvalues are obtained as follows:

$$\lambda_{1,2} = \frac{(1 + \chi\omega + \frac{\chi\varphi}{\triangle\tau} \pm \gamma)}{2},$$

$$\gamma = \sqrt{(1 + \chi\omega + \frac{\chi\varphi}{\triangle\tau})^2 - 4\chi\omega}$$

with $\lambda_1 \geq \lambda_2$. The explicit form of the recurrence relation (29) is then given by

$$Y_i^1(t) = r_1 + r_2\lambda_1^t + r_3\lambda_2^t$$

where r_1, r_2, and r_3 are constants determined by the initial conditions of the system. From updated velocity

$$V_i^1(t+1) = \frac{Y_i^1(t+1) - Y_i(t)}{\triangle\tau} \tag{40}$$

result in

$$V_i^1(t+1) = \frac{r_2(\lambda_1^{t+1} - \lambda_1^t) + r_3(\lambda_2^{t+1} - \lambda_2^t)}{\triangle\tau}$$

$$V_i^1(t+1) = (r_2\frac{\lambda_1 - 1}{\triangle\tau})\lambda_1^t + (r_3\frac{\lambda_2 - 1}{\triangle\tau})\lambda_2^t$$

$k_1 = \frac{r_1(\lambda_1 - 1)}{\triangle\tau}$ and $k_2 = \frac{r_2(\lambda_2 - 1)}{\triangle\tau}$

$$\lim_{t\to\infty} V_i^1(t+1) = \lim_{t\to\infty} k_1\lambda_1^t + \lim_{t\to\infty} k_2\lambda_2^t$$

$$\lim_{t\to\infty} X_i^1(t+1) = \begin{cases} \lim\limits_{t\to\infty} x_i^1(t) \ \ if\ max(||\lambda_1||, ||\lambda_2||) < 1, \\ (k_1\ or\ k_2\ or\ k_1 + k_2) + \lim\limits_{t\to\infty} x_i^1(t)\ if \\ \quad max(||\lambda_1||,\ ||\lambda_2||) = 1 \end{cases} \tag{41}$$

(41) implies that if the CSPSO algorithm is convergent, then velocity of the particles will decrease to zero or stay unchanged until the end of the iteration.

3.3 Constriction factor and its impact

When the PSO algorithm is run without controlling the velocity, the system explodes after a few iterations. To control the convergence properties of a particle swarm system, an important model having constriction factor and ω together is shown below:

$$\begin{aligned} V_i^{t+1} = \chi * \{\omega V_i^t + \varphi_1(P_i^t - x_i^t) + \\ \varphi_2(g_i^t - x_i^t)\}, \end{aligned} \tag{42}$$

$$2\kappa = \chi \mid 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \mid,$$

$\varphi_1 + \varphi_2 = \varphi \geq 4,\ \ 0 \leq \kappa \leq 1.$

Under these assumption conditions, the particle's trajectory in the CSPSO system is stable [6].

$$\begin{aligned} \omega^{t+1} = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{t_{max}}\right)t, \\ \omega_{max} > \omega_{min} \end{aligned} \tag{43}$$

where, ω_{max} and ω_{min} are the predefined initial and final values of the inertia weight, respectively, t_{max} is the maximum iteration number, and t is the current iteration number for a linearly decreasing inertia weight scheme.

3.4 Global convergence analysis of QBCSPSO

A sequence generated by the iterative PSO algorithm converges to a solution point. Several PSO variants were proposed to enhance convergence performance of PSO [5, 24], which combines quantum results with CSPSO, denoted as QBCSPSO. In this subsection, the global convergence of QBCSPSO is investigated.

From the Monte Carlo method, the current velocity for the position x_i^{t+1} of the d th element of the i th particle at the $(t+1)$ th iteration in QBCSPSO can be obtained using the following formula:

$$V_i^{t+1} = Y_i^t \pm L_i^t \, ln(u_i^{-(t+1)}), \quad u_i^{t+1} \sim U(0,1) \tag{44}$$

where $U_i^t \in (0,\ 1)$. Referring to Sun et al. [18, 24], where δ the (wave) function

$$\delta(Y_i^{t+1}) = \frac{1}{\sqrt{L_i^t}} \, exp(-Y_i^{t+1}/L_i^t)$$

with $Y_i^{t+1} = \mid x_i^{t+1} - P_i^t \mid$, and

the characteristic length L_i^t is obtained by

$$L_i^t = \gamma |x_i^t - C^t| \tag{45}$$

the term C^t used in (45) is $C^t = \frac{1}{K} \sum_{i=1}^{K} P_i^t$. [24].
The contraction-expansion coefficient γ can be adjusted to balance the trade-off between global and local exploration ability of the particles during the optimization process for two main purposes [38]:

- a larger γ value enables particles to have a stronger exploration ability but a less exploitation ability.
- a smaller γ allows particles a more precise exploitation ability.

Notice that in this article, most of the (1)-(45) represent velocities or positions or both of them.

4 Results and discussions

To demonstrate the working of the CSPSO algorithm, two well-known test functions in a global optimization were widely used in evaluating performance of evolutionary methods, and have the global minimum at the origin or very close to the origin. We compare the performance of PSO, SPSO, CPSO, and CSPSO.
Example 1. Unimodal function

$$min\, f(x_i) = \sum_{i=1}^{K} x_i^2 \quad Subject\ to - 10 \le x_i \le 10. \tag{46}$$

Example 2. Multi modal function

$$min\, f(x) = \sum_{i=1}^{K} -x_i sin(\sqrt{\mid x_i \mid}) \quad Subject\ to - 10 \le x_i \le 10. \tag{47}$$

In the experiments, inertia weight decreases from 0.9 to 0.4 and the generation stops when $E_i = \mid Fg^t(x_i) - Fp^t(x_i) \mid \le tolerance$ satisfied. Here, Fp is the function value of the best personal in current iteration and Fg denotes the global optimum and $c_1 = c_2 = 1.49$ and $c_1 = c_2 = 2$ are used in PSO and CSPSO, respectively.

For all algorithms, results are averaged over 100 independent runs and iterations while the population size is 50.

Following the recommendations of the original references, the best function value settings of some compared algorithms are summarized in Table 1.

The mean velocity v^{t+1} of (46) is shown using Table 1 and graphically (Figures 1–3) for the algorithms in Table 1. Figure 1 shows the convergence of PSO without controlling factor inertia weight exploded. One of the main limitations of PSO is that particles prematurely converge toward a local solution.

The evaluation results of the compared algorithms are shown in Figures 2, 3 for decreasing and increasing inertia weight, respectively. Figure 2 shows the evolution of inertial weight of the compared algorithms over the running time. The main disadvantage is that once the inertia weight is decreased, the

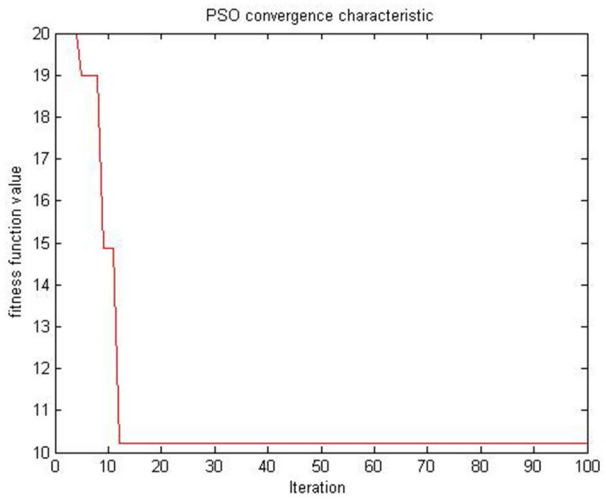

FIGURE 1
Basic PSO with no inertia weight for (46) on example 1.

TABLE 1 Comparison of algorithms on optimization test functions.

No	Best fun	Algorithm	Best run	Best variables	$\omega \in range$
1.	10.2213	Basic PSO	35	[0.3809, 1.4531, 3.1164]	
2.	9.3944	SPSO	4	[0.4381, 1.4566, 3.1052]	[0.9, 0.4]
3.	9.3945	SPSO	27	[0.4384, 1.4564, 3.1051]	[0.4, 0.9]
4.	9.3941	CSPSO	29	[0.4379, 1.4568, 3.1053]	[0.9, 0.4]
5.	9.3941	CSPSO	28	[0.4381, 1.4567, 3.1052]	[0.4, 0.9]

swarm loses its ability to search new areas [39]. Figure 3 shows the evolution of convergence characteristic for CSPSO based on inertial weight during the run. CSPSO can avoid premature convergence by performing efficient exploration that can help to find better solutions as the number of iterations increases and can avoid premature convergence by balancing exploration and exploitation. The algorithm CSPSO has shown fast convergence speed on unimodal functions.

In order to confirm the performance on multi-modal functions, we carry out a similar simulation by using (47).

The same set of parameters is assigned for all algorithms of Table 2 as in (46). Where in this function the number of local minima increases exponentially with the problem dimension. Its global optimum value is approximately −5.74, as we see from Table 2.

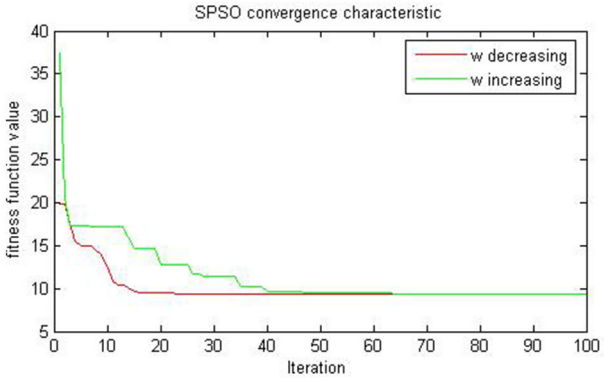

FIGURE 2
Evaluation of SPSO for (46) on example 1.

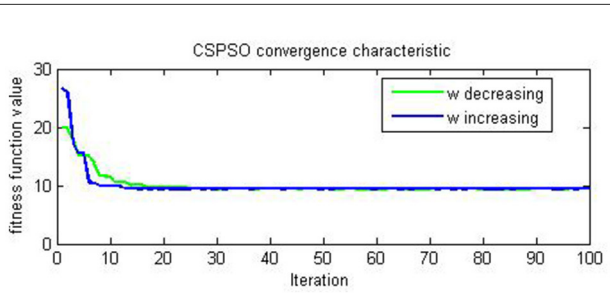

FIGURE 3
Evaluation of CSPSO for (46) on example 1.

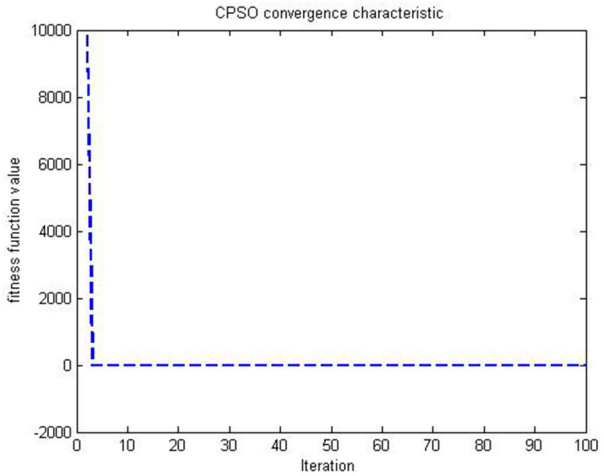

FIGURE 4
CPSO when $\omega = 1$ for (47) on example 2.

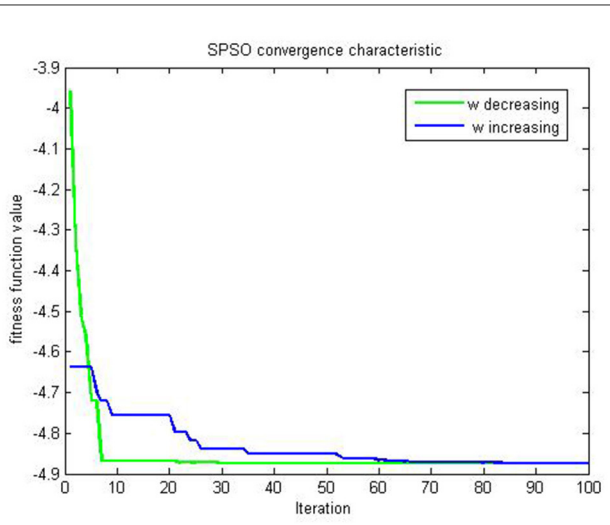

FIGURE 5
Evaluation of SPSO for (47) on example 2.

TABLE 2 **Comparison of algorithms on optimization test functions of multi-modal.**

No	Algorithm	Best fun	Best run	Best variables	$\omega \in range$
1.	CPSO	−6.12	01	[1.79, 0, 3.20]	$\omega = 1$
2.	SPSO	−6.73	32	[1.79, 0, 3.21]	[0.4, 0.9]
3.	SPSO	−8.133	89	[1.79, 0, 3.21]	[0.9, 0.4]
4.	CSPSO	−11.83	69	[5.23, 5.23, 5.23]	[0.9, 0.4]
5.	CSPSO	−5.74	39	[1.79, 0, 3.21]	[0.4, 0.9]

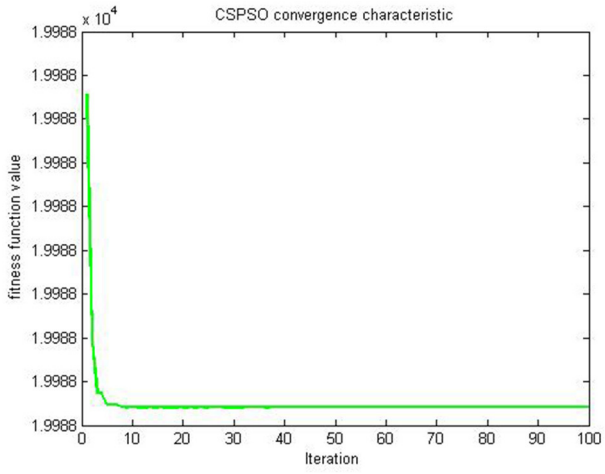

FIGURE 6
Evaluation of CSPSO for (47) on example 2.

Figures 4, 5 are simulations obtained from Table 2 results, and all the figures meet the objective of the CSPSO algorithm for the optimization problem given in (47) and its evaluation in Figure 6.

5 Conclusion

This article mainly concerns the convergence and stability analysis of the CSPSO algorithm and its performance improvement for different constriction coefficients. We first investigated the convergence of the SPSO algorithm by relating it to the Markov chain in which the stochastic process and Markov properties employ quantum behaviors to improve the global convergence and prove Markov chain transition probability, showing that the CSPSO algorithm converges to the global optimum in probability. We also compared the proposed algorithm with basic PSO, SPSO, and CPSO algorithms evaluating the optimal value (fitness value) based on the range of ω. The proposed algorithm is fast and efficient, and the run plans of CSPSO for ω linearly decreasing from 0.9 to 0.4 are easy to implement. The CSPSO algorithm performs better because it regenerated those results to minimize the test functions. On the other hand, the proposed heuristic algorithm did not seek solutions that minimized the delay time or cost function, and the adjustment process would be stopped if no ω was identified as regular. The CSPSO algorithm is verified to be a global convergent algorithm. These promising results motivate other researchers to apply CSPSO to solve optimization problems. And in the future we will make further investigations on convergence and stability of PSO variants.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

DT: Formal analysis, Writing—original draft, Software, Investigation, Project administration. TG: Supervision, Methodology, Validation, Project administration, Writing—review & editing. SL: Conceptualization, Supervision.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

Authors are grateful to the referees and handling editor for their constructive comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Kennedy J, Eberhart RC. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, (1995). p. 1942–1948.

2. Ahmed G. GadParticle swarm optimization algorithm and its applications: a systematic review. *Arch Comput Methods Eng.* (2022) 29:2531–61. doi: 10.1007/s11831-021-09694-4

3. Banks A, Vincent J, Anyakoha CA. A review of particle swarm optimization. Part I: background and development. *Nat Comput.* (2007) 6:467–84. doi: 10.1007/s11047-007-9049-5

4. AlRashidi MR, El-Hawary ME. A survey of particle swarm optimization applications in electric power systems. *IEEE Trans Evol Comput.* (2009) 14:913–8. doi: 10.1109/TEVC.2006.880326

5. Kumar A, Singh BK, Patro BD. Particle swarm optimization: a study of variants and their applications. *Int J Comput Applic*. (2016) 135:24–30. doi: 10.5120/ijca2016908406

6. Tarekegn D, Tilahun S, Gemechu T. A review on convergence analysis of particle swarm optimization. *Int J Swarm Intell Res*. (2023) 14:328092. doi: 10.4018/IJSIR.328092

7. Clerc M, Kennedy J. The particle swarm-explosion, stability and convergence in a multidimensional complex space. *IEEE Trans Evol Comput*. (2002) 6:58–73. doi: 10.1109/4235.985692

8. van den Bergh F, Engelbrecht AP. A study of particle swarm optimization particle trajectories. *Inform Sci*. (2006) 178:937–71. doi: 10.1016/j.ins.2005.02.003

9. Kadirkamanathan V, Selvarajah K, Fleming PJ. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans Evol Comput*. (2006) 10:245–55. doi: 10.1109/TEVC.2005.857077

10. Liu QF. Order-2 stability analysis of particle swarm optimization. *Evol Comput*. (2015) 23:187–216. doi: 10.1162/EVCO_a_00129

11. Dong WY, Zhang RR. Order-3 stability analysis of particle swarm optimization. *Inf Sci*. (2019) 503:508–20. doi: 10.1016/j.ins.2019.07.020

12. Jiang M, Luo YP, Yang SY. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Inform Process Lett*. (2007) 102:8–16. doi: 10.1016/j.ipl.2006.10.005

13. Poli R, Langdon WB. Markov chain models of bare-bones particle swarm optimizers. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. (2007). p. 142–149. doi: 10.1145/1276958.1276978

14. Cai ZQ, Huang H, Zheng ZH, Luo W. Convergence improvement of particle swarm optimization based on the expanding attaining-state set. *J Huazhong Univ Sci Technol*. (2009) 37:44–7.

15. Solis F, Wets R. Minimization by random search techniques. *Math Oper Res*. (1981) 6:19–30. doi: 10.1287/moor.6.1.19

16. van den Bergh F, Petrus Engelbrecht A. A convergence proof for the particle swarm. *Optimiser Fund Inform*. (2010) 105:341–74. doi: 10.3233/FI-2010-370

17. Dirk S, Carsten W. Runtime analysis of a binary particle swarm optimizer. *Theor Comput Sci*. (2010) 411:2084–100. doi: 10.1016/j.tcs.2010.03.002

18. Sun J, Wu X, Palade V, Fang W, Lai CH, Xu W. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Inf Sci*. (2012) 193:81–103. doi: 10.1016/j.ins.2012.01.005

19. Per KL, Carsten W. Finite first hitting time versus stochastic convergence in particle swarm optimization. *Adv Metaheur*. (2013) 53:1–20. doi: 10.1007/978-1-4614-6322-1_1

20. Yuan Q, Yin G. Analyzing convergence and rates of convergence of particle swarm optimization algorithms using stochastic approximation methods. *IEEE T Automat Contr*. (2015) 60:1760–73. doi: 10.1109/TAC.2014.2381454

21. Ren ZH, Wang J, Gao Y. The global convergence analysis of particle swarm optimization algorithm based on Markov chain. *Control Theory Applic*. (2011) 28:462–6.

22. Xu G, Yu G. On convergence analysis of particle swarm optimization algorithm. *J Comput Appl Math*. (2017) 333:65–73. doi: 10.1016/j.cam.2017.10.026

23. Feng P, Xiao-Ting LI, Qian ZH, Wei-Xing LI Qi GA. Analysis of standard particle swarm optimization algorithm based on Markov Chain. *Acta Autom Sinica*. (2013) 39:381–9. doi: 10.1016/S1874-1029(13)60037-3

24. Sun J, Feng B, Xu WB. Particle swarm optimization with particles having quantum behavior. In: *Proceedings of the 2004 Congress on Evolutionary Computation, CEC04,* Portland, USA (2004). p. 326–331.

25. Shi YH, Eberhart RC. Empirical study of particle swarm optimization. In: *Proceedings of the Congress on Evolutionary Computation*, (1999). p. 1945–1950.

26. Kennedy J. Bare bones particle swarms. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, IEEE (2003). p. 80–87.

27. Kennedy J, Mendes R. Population structure and particle swarm performance. In: *Proceedings of the Congress on Evolutionary Computation.* (2002). p. 1671–6.

28. Liang JJ, Suganthan PN. Dynamic multi-swarm particle swarm optimizer. In: *Proceedings of the IEEE Swarm Intelligence Symposium.* (2005). p. 124–129.

29. Parrott D, Li X. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Trans Evol Comput*. (2006) 10:440–58. doi: 10.1109/TEVC.2005.859468

30. Brits R, Engelbrecht AP, van den Bergh F. Locating multiple optima using particle swarm optimization. *Appl Math Comput*. (2007) 189:1859–83. doi: 10.1016/j.amc.2006.12.066

31. Zhao X. A perturbed particle swarm algorithm for numerical optimization. *Appl Soft Comput*. (2010) 10:119–24. doi: 10.1016/j.asoc.2009.06.010

32. Tang Y, Wang ZD, Fang JA. Feedback learning particle swarm optimization. *Appl Soft Comput*. (2011) 11:4713–25. doi: 10.1016/j.asoc.2011.07.012

33. Wang HF, Moon I, Yang SX, Wang DW. A memetic particle swarm optimization algorithm for multimodal optimization problems. *Inform Sci*. (2012) 197:38–52. doi: 10.1016/j.ins.2012.02.016

34. Huang H, Qin H, Hao Z, Lim A. Example-based learning particle swarm optimization for continuous optimization. *Inform Sci*. (2012) 182:125–38. doi: 10.1016/j.ins.2010.10.018

35. Lawler GF. Introduction to Stochastic Processes, second edition. London: Chapman and Hall/CRC Press. (2006).

36. Xiao XY, Yin HW. Moment convergence rates in the law of logarithm for moving average process under dependence. *Stochastics*. (2014) 86:1–15. doi: 10.1080/17442508.2012.748057

37. Hu D, Qiu X, Liu Y, Zhou X. Probabilistic convergence analysis of the stochastic particle swarm optimization model without the stagnation assumption. *Inf Sci*. (2021) 547:996–1007. doi: 10.1016/j.ins.2020.08.072

38. Chen J, Xin B, Peng Z, Dou L, Zhang J. Optimal contraction theorem for explorationexploitation tradeoff in search and optimization. *IEEE Trans Syst Man Cybern Part A*. (2009) 39:680–91. doi: 10.1109/TSMCA.2009.2012436

39. Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH. Prairie dog optimization algorithm. *Neural Comput Applic*. (2022) 34:20017–65. doi: 10.1007/s00521-022-07530-9

40. Lee J, Mukerji T. *Probabilistic Particle Swarm Optimization (Pro]PSO) for Using Prior Information and Hierarchical Parameters*. Department of Energy Resources Engineering, Stanford University (2014).

41. Schmitt M, Wanka R. Particle swarm optimization almost surely finds local optima. *Theor Comput Sci*. (2014) 561:57–72. doi: 10.1016/j.tcs.2014.05.017

Appendix

Matlab codes

```
tic
clc
clear all
close all
rng default
LB=[0 0 0]; UB=[10 10 10];
m=3; n=50;
wmin=0.9; wmax=0.4; c1=2; c2=2;
maxite=100; maxrun=100;
for run=1:maxrun
run
for i=1:n
for j=1:m
x0(i,j)=round(LB(j)+(UB(j)-LB(j))*rand());
end
end
x=x0; v=0.1*x0;
for i=1:n
f0(i,1)=ofun(x0(i,:));
end
pbest=x0;
[fmin0,index0]=min(f0);
gbest=x0(index0,:);
ite=1;
tolerance=1;
rho=0.9;
while ite <= maxite && tolerance > 10^-12
w =wmax-(wmax-wmin)*ite/maxite;
kappa=1; phi1=2.05; phi2=2.05;
phi=phi1+phi2;
chi = 2 * kappa/abs(2 - phi - sqrt(phi^2 - 4 * phi));
a=1/w;
for i=1:n
for j=1:m
v(i,j)=chi*[w*v(i,j)+c1*rand()*(pbest(i,j)-
x(i,j))+c2*rand()*(gbest(1,j)-x(i,j))];
end
end
for i=1:n
for j=1:m
x(i,j)=x(i,j)+v(i,j);
end
end
for i=1:n
for j=1:m
if x(i,j)<LB(j)
x(i,j)=LB(j);
elseif x(i,j)>UB(j)
x(i,j)=UB(j);
end
end
end
for i=1:n
f(i,1)=ofun(x(i,:));
end
for i=1:n
if f(i,1)<f0(i,1)
pbest(i,:)=x(i,:);
f0(i,1)=f(i,1);
end
end
[fmin,index]=min(f0);
ffmin(ite,run)=fmin;
ffite(run)=ite;
if fmin<fmin0
gbest=pbest(index,:);
fmin0=fmin;
end
if ite>100;
tolerance=abs(ffmin(ite-100,run)-fmin0);
end
if ite==1;
disp(sprintf('Iteration Best particle objective fun'));
end
disp(sprintf('
ite=ite+1;
end
gbest;
fvalue=-x(1)*sin(sqrt(abs(x(1))))-x(2)*sin(sqrt(abs(x(2))))-
x(3)*sin(sqrt(abs(x(3))));
fff(run)=fvalue;
rgbest(run,:)=gbest;
disp(sprintf('———'));
end
disp(sprintf(''));
disp(sprintf('**********'));
disp(sprintf('Final Results——'));
[bestfun, bestrun] = min(fff)
best_variables = rgbest(bestrun, :)
disp(sprintf('**********'));
toc
plot(ffmin(1:ffite(bestrun),bestrun),'–b','linewidth',2);
xlabel('Iteration ');
ylabel('fitness function value');
title('CSPSO convergence characteristic')
```