On Convergence and Parameter Selection of an Improved Particle Swarm Optimization

Xin Chen and Yangmin Li*

Abstract: This paper proposes an improved particle swarm optimization named PSO with Controllable Random Exploration Velocity (PSO-CREV) behaving an additional exploration behavior. Different from other improvements on PSO, the updating principle of PSO-CREV is constructed in terms of stochastic approximation diagram. Hence a stochastic velocity independent on cognitive and social components of PSO can be added to the updating principle, so that particles have strong exploration ability than those of conventional PSO. The conditions and main behaviors of PSO-CREV are described. Two properties in terms of "divergence before convergence" and "controllable exploration behavior" are presented, which promote the performance of PSO-CREV. An experimental method based on a complex test function is proposed by which the proper parameters of PSO-CREV used in practice are figured out, which guarantees the high exploration ability, as well as the convergence rate is concerned. The benchmarks and applications on FCRNN training verify the improvements brought by PSO-CREV.

Keywords: Lyapunov theory, PSO-CREV, stochastic approximation, supermartingale convergence.

1. INTRODUCTION

Particle swarm optimization (PSO) is inspired from studies of social behavior among ants and birds [1,2]. Now it has been widely used to solve non-linear and multi-objective problems, such as optimization of weights of NN, electrical utility, computer games, and mobile robots, etc. [3-6].

As a recursive algorithm, the PSO algorithm simulates social behavior among individuals (particles) "flying" through a multidimensional search space, where each particle represents a point at the intersection of all search dimensions. Considering movements of particles, PSO is always modeled as second order systems in continuous-time version [7] or discrete-time versions [8-11]. Because these researches are based on a kind of reduced system [8], in which all stochastic factors are simplified, it is difficult to analyze the stochastic behavior of PSO.

The first significant improvement is the inertia weight [12], which results in faster convergence. Another significant improvement about PSO convergence is the constrict coefficient [8], which ensures the convergence of PSO. In fact, almost all improvements on PSO are based on these two basic improvements.

Normally the improvements on PSO are referred to two aspects: promoting convergence rate and avoiding local minima or prematurity, especially the latter one. Obviously to promote exploration ability of particles is the direct way to drive particles out of local minima. Hence many improvements are proposed to improve PSOs exploration ability. For example, in cooperative PSO [13], several swarms work together to optimize different components of the solution vector cooperatively, so that there are more chances to find the global best solution. On the other hand, in order to drive particles to unknown subspace, a kind of mutation, or similar operation, is introduced to PSO, such as MPSO [14] and GCPSO [15], where some particles are selected to change their positions randomly, so that they may reach some unknown solution space. Moreover to improve search ability, hybrid PSO algorithms with other optimization techniques, such as GAs [3] and chaotic search [16], etc., are developed.

These improvements about PSO have a character that no matter how additional operations are employed, all of them use the basic updating principles with inertia weight or constrict coefficient to adjust particles positions. In this paper, we propose a novel

Manuscript received May 25, 2006; accepted March 5, 2008. Recommended by Editor Young-Hoon Joo.

Xin Chen is with the School of Information Science and Engineering, Central South University, Changsha, Hunan, China (e-mail: chenxin@mail.csu.edu.cn).

Yangmin Li is with the Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira, Taipa, Macao SAR, China (e-mail: ymli@umac.mo).

* Corresponding author.

PSO algorithm which results from the direct modification on conventional PSO structure, where two major modifications are introduced.

On one hand, a decreasing proportional coefficient is introduced to PSO. Such a decreasing coefficient results in several modifications on PSO structure, so that the new PSO is convergent.

On another hand, a random exploration velocity is introduced into the new PSO. Since this random velocity plays a very important role, the new PSO algorithm is named PSO with Controllable Random Exploration Velocity (PSO-CREV).

The rest of this paper is organized as follows. In Section 2, we introduce the motivation of PSO-CREV diagram. And the description on PSO-CREV is introduced in form of a theorem. In Section 3 convergence of PSO-CREV is proved using the theory of stochastic approximation [17,18]. And based on the proof, two important properties induced by stochastic components of PSO-CREV are summarized. To choose proper values of parameters of PSO-CREV, an experiment is employed in Section 4 where parameters are determined according to experimental results. Experimental settings for the benchmarks and simulation results are introduced in Section 5 and a practical implementation of PSO-CREV in NN training is presented in Section 6. Finally conclusions are drawn in Section 7.

2. MOTIVATION AND DEFINITION OF PSO-CREV

2.1. Concept and constraints of conventional PSO

If let M denote the size of the swarm, the current position of particle i is denoted by $X_i = [X_{i1}\ X_{i2}\ \cdots\ X_{iD}]^T$, $i = 1, 2, \cdots, M$, where D is the dimension of the solution space, and its current velocity is denoted by v_i. Then the updating principle is expressed as

$$\begin{aligned} v_{id}(n+1) &= v_{id}(n) + c_1 r_{1d}(n)[P_{id}^d(n) - X_{id}(n)] \\ &\quad + c_2 r_{2d}(n)[P_{id}^g(n) - X_{id}(n)], \end{aligned} \tag{1}$$

$$X_{id}(n+1) = X_{id}(n) + v_{id}(n+1), \tag{2}$$

where $d = 1, \cdots, D$, $r_{1d} \sim U(0,1)$ and $r_{2d} \sim U(0,1)$ represent the two random numbers within [0,1]; c_{1i} and c_{2i} represent the acceleration coefficients; $P_i^d(n)$ represents the best position found by particle i so far, $P_i^g(n)$ represents the global best position found by particle i's neighborhood.

Two random variables r_{1d} and r_{2d} are employed to realize exploration. But the intension of exploration behavior is totally determined by the decreasing rate of cognitive and social components. Thus if a swarm converges quickly to a certain position maybe not the global best solution, particles also give up attempts for exploration. Then particles are stagnated at local minima.

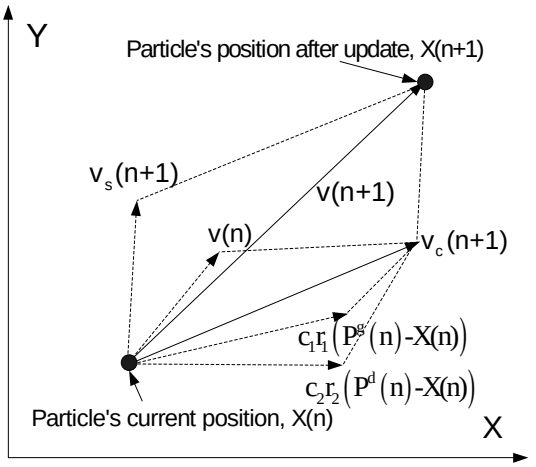

Fig. 1. The concept of PSO-CREV, where a stochastic velocity $v_s(n+1)$ adjusts the search direction of the particle.

To overcome this disadvantage, an additional exploration behavior looks very helpful to pull particles out of local minima. Just as shown in Fig. 1, where $v_c(n+1)$ represents the velocity of a particle after conventional updating principle, i.e., the velocity determined by $P^d(n)$ and $P^g(n)$. There is a random velocity denoted by $v_s(n+1)$ added to $v_c(n+1)$, so that a new velocity $v(n+1)$ is obtained, which drives particle to search other direction. Even if $P^d(n) - X(n)$ and $P^g(n) - X(n)$ decrease quickly, the particle can maintain exploration ability.

In a sense, this velocity is viewed as a disturbance to the system, which must affect the convergence of the system. To ensure convergence of the system, an additional operation suppressing this random variable (denoted by $\varepsilon(n)$ in this paper) is necessary. In the following subsection, the motivation of PSO-CREV is introduced, which is based on a modified reduced system.

2.2. Modifications on the Conventional PSO

Modification 1: A decreasing $\varepsilon(n)$ is added into the expression of updating velocity, which goes to zero as $n \to \infty$.

If let $Y(n) = X(n) - P$, where P represents the best solution found so far, the reduced system, or implicit representation (IR) [8] is expressed as follow, where the subscript i is omitted for brevity.

$$\begin{aligned} v(n+1) &= \varepsilon v(n) - \varepsilon\phi Y(n) \\ Y(n+1) &= \varepsilon v(n) + (1 - \varepsilon\phi) Y(n)). \end{aligned} \tag{3}$$

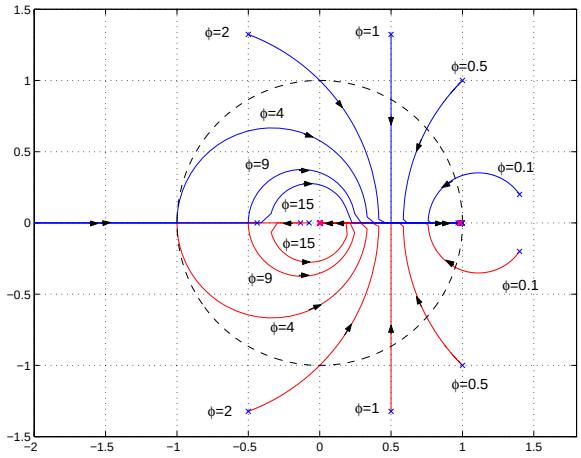

Fig. 2. Root locus plot with different ϕ as $\varepsilon(n) \to 0$.

It is easy to obtain the characteristic polynomial of the system as follows

$$Z^2 + (\varepsilon\phi - 1 - \varepsilon)Z + \varepsilon = 0. \tag{4}$$

Hence the eigenvalues are expressed as

$$e_{1,2} = \frac{1 + \varepsilon - \varepsilon\phi \pm \sqrt{(1 + \varepsilon - \varepsilon\phi)^2 - 4\varepsilon}}{2}. \tag{5}$$

Consequently from the root locus of the reduced system shown in Fig. 2, it is observed that no matter what ϕ equals, as $\varepsilon(n)$ is approaching to 0, the roots go into a unit circle in form of complex roots, and finally converge to two real roots 0 and 1.

According to the stability theorem of discrete system, along with decreasing $\varepsilon(n)$, one of two roots will converge to 1, so that the system maintains sustained oscillation ultimately with frequency 1. That means the reduced system is nonconvergent. To overcome this drawback, the second modification is introduced as follows.

Modification 2: A positive coefficient α is added to the expression of updating position, which is less than 1. Hence the reduced system is changed into

$$\begin{aligned} v(n+1) &= \varepsilon v(n) - \varepsilon\phi Y(n), \\ Y(n+1) &= \varepsilon v(n) + (\alpha - \varepsilon\phi)Y(n)). \end{aligned} \tag{6}$$

Consequently both roots converge to α and 0, so that the reduced system is convergent.

Now extend the reduced system (6) to the updating principle of PSO, P is substituted by $\frac{\phi_1 P^d + \phi_2 P^g}{\phi_1 + \phi_2}$, where $\phi_1 = c_1 r_1$ and $\phi_2 = c_2 r_2$, ε is substituted by $\varepsilon(n)$, and ϕ is substituted by $\phi_1 + \phi_2$. Hence (6) is expressed as

$$\begin{aligned} v(n+1) &= \varepsilon(n)v(n) + c_1 r_1 (P^d - X(n)) \\ &\quad + c_2 r_2 (P^g - X(n)), \\ X(n+1) &= \alpha X(n) + v(n+1) + \frac{1-\alpha}{\phi_1 + \phi_2}(\phi_1 P^d + \phi_2 P^g). \end{aligned}$$

Comparing with the conventional PSO algorithm, there exists an additional term on the right hand of position updating. Based on above analysis, a complete definition of the novel PSO algorithm is expressed as the following theorem.

2.3. Definition of PSO with Controllable Random Exploration Velocity (PSO-CREV)

Theorem 1: A *PSO with Controllable Random Exploration Velocity* (PSO-CREV) is described as follows. Let $\mathcal{F}(n)$ be a sequence of sub-σ-algebra of $\mathcal{F}$ such that $\mathcal{F}(n) \subset \mathcal{F}(n+1)$, for all n. For a swarm including M particles, the position of particle i is defined as $X_i = [x_{i1} \quad x_{i2} \quad \cdots \quad x_{iD}]^T$, where D represents the dimension of swarm space. The updating principle for individual particle is defined as

$$\begin{aligned} v_{id}(n+1) &= \varepsilon(n)\Big[v_{id}(n) + c_1 r_{1id}(n)(P_{id}^d(n) - X_{id}(n)) \\ &\quad + c_2 r_{2id}(n)(P_{id}^g(n) - X_{id}(n)) + \xi_{id}(n)\Big], \\ X_{id}(n+1) &= \alpha X_{id}(n) + v_{id}(n+1) \\ &\quad + \frac{1-\alpha}{\phi_{id}(n)}(c_1 r_{1id}(n)P_{id}^d(n) + c_2 r_{2id}(n)P_{id}^g(n)), \end{aligned} \tag{7}$$

where $d = 1, \cdots, D$, c_1 and c_2 are positive constants; $r_{1id}(n)$ and $r_{2id}(n)$ are $\mathcal{F}(n)$-measurable random variables; $P_i^d(n)$ represents the best position that particle i has found so far, which is of the form $P_i^d(n) = \text{argmin}_{k \le n} F(X_i(k))$, where $F(\cdot)$ represents a fitness function to be decreased; $P_i^g(n)$ represents the best position found by particle i's neighborhood, which is of the form $P_i^g(n) = \arg\min_{j \in \Pi_i} F(X_j(n))$; $\phi_i(n) = \phi_{1i}(n) + \phi_{2i}(n)$, where $\phi_{1i}(n) = c_1 r_{1i}(n)$, $\phi_{2i}(n) = c_2 r_{2i}(n)$.

Suppose the following assumptions hold:

1) $\xi_i(n)$ is a bounded random variable with continuous uniform distribution. It has a constant expectation denoted by $\Xi_i = \mathbf{E}\xi_i(n)$;

2) $\varepsilon(n) \to 0$ with n increasing, and $\sum_{n=1}^{\infty} \varepsilon(n) = \infty$;

3) $0 < \alpha < 1$;

4) $r_{1id}(n)$ and $r_{2id}(n)$ are independent random variables satisfying continuous uniform distribution in [0,1], or $r_{1id} \sim U(0,1)$ and $r_{2id} \sim U(0,1)$. And denote $\Phi_{1i} = \mathbf{E}\phi_{1i}(n)$ and $\Phi_{2i} = \mathbf{E}\phi_{2i}(n)$ respectively.

Then swarm must converge with probability one. Let $P^* = \inf_{\lambda \in (\mathbf{R}^D)} F(\lambda)$ represent the global optimal solution. Then swarm converges to P^* if $\lim_{n\to\infty} P_i^d(n) \to P^*$ and $\lim_{n\to\infty} P_i^g(n) \to P^*$.

Because the stochastic properties are all ignored in the reduced system, we apply the theory about stochastic approximation to prove the convergence of the improved PSO paradigm.

3. CONVERGENCE AND PROPERTIES OF PSO-CREV

3.1. Convergence of PSO-CREV

A new relative position is defined as $Y(n) = X(n) - P^*$. Substitute it into (7), and change (7) into

$$\begin{aligned}
&v(n+1)\\
&= v(n) + [-(1-\varepsilon(n))v(n) - \varepsilon(n)\phi(n)Y(n)\\
&\quad + \varepsilon(n)(\phi_1(n)Q^d(n) + \phi_2(n)Q^g(n)) + \varepsilon(n)\xi(n)],\\
&Y(n+1)\\
&= Y(n) + (\alpha-1)Y(n) + v(n+1)\\
&\quad + \frac{1-\alpha}{\phi(n)}(\phi_1(n)Q^d(n) + \phi_2(n)Q^g(n))\\
&= Y(n) + \Big[-(1-\alpha+\varepsilon(n)\phi)Y(n) + \varepsilon(n)v(n)\\
&\quad + \frac{1-\alpha+\varepsilon(n)\phi(n)}{\phi(n)}(\phi_1(n)Q^d(n) + \phi_2(n)Q^g(n))\Big],
\end{aligned} \tag{8}$$

where $Q^d(n) = P^d(n) - P^*$, $Q^g(n) = P^g(n) - P^*$. According to the updating of $P^d(n)$ and $P^g(n)$, $\|Q^d(n)\|$ and $\|Q^g(n)\|$ are nonincreasing over iterations, if P^* denotes the global best solution.

If let $Q^r(n) = \frac{1}{\phi(n)}[\phi_1(n)Q^d(n) + \phi_2(n)Q^g(n)]$, $\mathbf{E}\left[\|Q^r(n)\| \mid F_n\right]$ must be nonincreasing over iterations. Let

$$\theta(n) = [v(n) \quad Z(n)]^T = [v(n) \quad Y(n) - \mathbf{E}_n Q^r(n)]^T,$$

where $\mathbf{E}_n$ denotes the expectation conditioned on the σ-algebra F_n. Then it follows that

$$\theta(n+1) = \theta(n) + \varepsilon(n)H(n), \tag{9}$$

where $H(n) = \frac{1}{\varepsilon(n)}[h_1 \quad h_2]^T$, where

$$\begin{aligned}
h_1 &= -(1-\varepsilon(n))v(n) - \varepsilon(n)\phi(n)Z(n)\\
&\quad + \varepsilon(n)\phi(n)[Q^r(n) - \mathbf{E}_n Q^r(n)] + \varepsilon(n)\xi(n),\\
h_2 &= -(1-\alpha+\varepsilon(n)\phi(n))Z(n) + \varepsilon(n)v(n)\\
&\quad + (1-\alpha+\varepsilon(n)\phi(n))\cdot[Q^r(n) - \mathbf{E}_n Q^r(n)].
\end{aligned}$$

A Lyapunov function candidate is defined as

$$L(\theta(n)) = \frac{1}{2}\theta^T \begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} \theta = \frac{1}{2}(v^2(n) + \Phi Z^2(n)), \tag{10}$$

where $\Phi = \mathbf{E}(\phi)$.

3.1.1 Properties on the derivative of Lyapunov function

Let $\gamma(n) = \mathbf{E}_n H(n)$. If $\mathbf{E}\,|H(n)| < \infty$, we have

$$\gamma(n) = \frac{1}{\varepsilon(n)}\begin{bmatrix} -(1-\varepsilon(n))v(n) - \varepsilon(n)\Phi Z(n) + \varepsilon(n)\Xi \\ -(1-\alpha+\varepsilon(n)\Phi)Z(n) + \varepsilon(n)v(n) \end{bmatrix}. \tag{11}$$

Using a truncated Taylor series expansion, we have

$$\begin{aligned}
E_n L\big(\theta(n+1)\big) - L\big(\theta(n)\big) &= \varepsilon(n) L'_\theta\big(\theta(n)\big)\gamma(\theta)\\
&\quad + E_n\left(\varepsilon^2(n) H(n)^T \begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} H(n)\right).
\end{aligned} \tag{12}$$

Calculating the first term on the right side of (12) yields

$$\begin{aligned}
&L_\theta{}^T(\theta(n)\gamma(n)\\
&= \frac{1}{\varepsilon(n)}\mathbf{E}_n\{-v^T(n)[(1-\varepsilon(n))v(n) + \varepsilon(n)\Phi Z(n) - \varepsilon(n)\Xi]\\
&\quad - \Phi Z^T(n)[(1-\alpha+\varepsilon(n)\Phi)Z(n) - \varepsilon(n)v(n)]\}\\
&= -\tfrac{1}{\varepsilon(n)}[(1-\varepsilon(n))\|v(n)\|^2 + \Phi(1-\alpha+\varepsilon(n)\Phi)\|Z(n)\|^2\\
&\quad - \Xi v^T(n)\varepsilon(n)].
\end{aligned} \tag{13}$$

Let $(\tilde{H}(n))^2$ denote $H(n)^T \begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} H(n)$. After some calculations, we have

$$\begin{aligned}
&\mathbf{E}\left[\|\tilde{H}(n)\| F(n)\right]^2\\
&= \frac{1}{\varepsilon^2(n)}\mathbf{E}_n\{a_1(n)\|v(n)\|^2 + a_2(n)\|Z(n)\|^2\\
&\quad + a_3(n)\|Q^r(n) - \mathbf{E}_n Q^r(n)\|^2 + (\varepsilon(n)\xi(n))^2\\
&\quad + a_4(n)v^T(n)Z(n) + a_5(n)[Q^r(n) - \mathbf{E}_n Q^r(n)]\\
&\quad + a_6(n)\xi(n)\},
\end{aligned} \tag{14}$$

where

$$
\begin{aligned}
a_1(n) &= (1-\varepsilon(n))^2 + \Phi\varepsilon^2(n),\\
a_2(n) &= \Phi(1-\alpha+\varepsilon(n)\phi(n))^2 + (\varepsilon(n)\phi(n))^2,\\
a_3(n) &= \Phi(1-\alpha+\varepsilon(n)\phi(n))^2 + (\varepsilon(n)\phi(n))^2,\\
a_4(n) &= 2\varepsilon(n)[\phi(n)(1-\varepsilon(n)) - \Phi(1-\alpha+\varepsilon(n)\phi(n))],\\
a_5(n) &= 2[\Phi\varepsilon(n)(1-\alpha+\varepsilon(n)\phi(n))\\
&\quad -\varepsilon(n)\phi(n)(1-\varepsilon(n))]v^T(n)\\
&\quad -2[\Phi(1-\alpha+\varepsilon(n)\phi(n))^2 + \varepsilon^2(n)\phi^2(n)]Z^T(n),\\
a_6(n) &= -2\varepsilon(n)\{(1-\varepsilon(n))v^T(n) + \varepsilon(n)\phi(n)Z^T(n)\\
&\quad -\varepsilon(n)\phi(n)[Q^r(n) - \mathbf{E}_n Q^r(n)]^T\}.
\end{aligned}
$$

Substituting (13) and (14) into (12), and using inequality $\|uv\| \le \frac{1}{2}(\|u\|^2 + \|v\|^2)$, we obtain

$$
\begin{aligned}
&\mathbf{E}_n L(\theta(n+1)) - L(\theta(n))\\
&\quad \le b_1(n)\|v(n)\|^2 + b_2(n)\|Z(n)\|^2\\
&\quad\quad + \mathbf{E}_n\left[b_3(n)\left\|Q^r(n) - E_n Q^r(n)\right\|^2\right] \qquad (15)\\
&\quad\quad + \mathbf{E}_n\left[b_4(n)(Q^r(n) - E_n Q^r(n))\right]\\
&\quad\quad + b_5(n)\mathbf{E}_n\|\xi(n)\|^2,
\end{aligned}
$$

where

$$
\begin{aligned}
b_1(n) &= -\tfrac{1}{2}\varepsilon(n) + \varepsilon^2(n)(2+\Phi) + \Phi^2\varepsilon^{\frac{3}{2}}(n),\\
b_2(n) &= -\Phi\alpha(1-\alpha) - \varepsilon(n)[\alpha\Phi - 2\varepsilon(n)E_n[\phi(n)]^2]\\
&\quad + \sqrt{\varepsilon(n)}[\alpha - (1+\Phi)\varepsilon(n)]^2,\\
b_3(n) &= \Phi(1-\alpha+\varepsilon(n)\phi(n))^2 + (\varepsilon(n)\phi(n))^2,\\
b_4(n) &= 2[\Phi\varepsilon(n)(1-\alpha+\varepsilon(n)\phi(n))\\
&\quad -\varepsilon(n)\phi(n)(1-\varepsilon(n))]v^T(n)\\
&\quad -2[\Phi(1-\alpha+\varepsilon(n)\phi(n))^2\\
&\quad +\varepsilon^2(n)\phi^2(n)]Z^T(n),\\
b_5(n) &= \tfrac{1}{2}\varepsilon(n) + 2\varepsilon^2(n).
\end{aligned}
$$

Obviously at beginning, if $\varepsilon(n)$ is large enough, $b_1(n)$ and $b_2(n)$ must be positive so that $\mathbf{E}_n L(\theta(n+1)) - L(\theta(n)) \ge 0$. Since $\varepsilon^2(n)$ and $\varepsilon^{3/2}(n)$ decrease faster than $\varepsilon(n)$, when n is large enough, $b_1(n)$ and $b_2(n)$ are negative. Along with $n \to \infty$, $\mathbf{E}_n\left[b_4(n)(Q^r(n) - \mathbf{E}_n Q^r(n))\right] \to 0$, and $b_5\mathbf{E}_n[\xi(n)]^2 \to 0$. That means that there exists $N_k < \infty$ such that when $n > N_k$, there is a positive non-decreasing function $k(\theta(n))$ to satisfy

$$
\begin{aligned}
&\mathbf{E}_n L(\theta(n+1)) - L(\theta(n))\\
&\quad \le -k(\theta(n)) + \mathbf{E}_n\left[b_3(n)\left\|Q^r(n) - E_n Q^r(n)\right\|^2\right]. \qquad (16)
\end{aligned}
$$

For a large n, (16) implies that the right side of (16) is negative outside of a neighborhood of the set $\left\{\theta \mid k(\theta(t)) \le \mathbf{E}_n\left[b_3(n)\left\|Q^r(t) - \mathbf{E}_n Q^r(t)\right\|^2\right]\right\}$. Outside such decreasing neighborhood, $L(\theta(n))$ has the supermartingale property. Then supermartingale convergence theorem implies that neighborhood of the set $\left\{\theta \mid k(\theta(t)) = \mathbf{E}_n\left[b_3(n)\left\|Q^r(t) - \mathbf{E}_n Q^r(t)\right\|^2\right]\right\}$ is recurrent, that is, $\theta(n)$ returns to it infinitely often with probability one. Since $\mathbf{E}_n\left[b_3(n)\left\|Q^r(t) - \mathbf{E}_n Q^r(t)\right\|^2\right] \to \Phi(1-\alpha)\mathbf{E}_n\left\|Q^r(t) - \mathbf{E}_n Q^r(t)\right\|^2$ as $n \to \infty$, $\theta(n)$ returns to neighborhood of $\{\theta \mid k(\theta(t)) = \Phi(1-\alpha)\mathbf{E}_n\left\|Q^r(t) - \mathbf{E}_n Q^r(t)\right\|^2\}$ infinitely often as $n \to \infty$.

3.1.2 Proof on bound of $E\|H(n)\|^2$

For $n \le N_k$, where N_k is defined as above, the proof of the bound of $\mathbf{E}\|H(n)\|^2$ is very similar to that of Lemma 5.4.1 in [18]. From (13) and (14), we observe that if $\varepsilon(n) < \infty$, $\mathbf{E}\|H(n)\|^2$ and $\mathbf{E}[L_\theta{}^T(\theta(n)\gamma(n))]$ are growing at most as $O(|\theta|^2)$. Therefore there are two positive constants K_1 and K_2 such that

$$
\mathbf{E}\|H(n)\|^2 + \mathbf{E}[L_\theta{}^T(\theta(n)\gamma(n))] \le K_1 L(\theta(n)) + K_2. \qquad (17)
$$

Firstly considering that $\mathbf{E}L(\theta(0)) < \infty$ and (17), we know that $\mathbf{E}\|H(0)\|^2 < \infty$. Suppose that $\mathbf{E}L(\theta(n)) \le \infty$ for some n. Then $\mathbf{E}\|H(n)\|^2 < \infty$. Using truncated Taylor series expansion shown in (12), (17) implies that there is a real K_3 such that the right side of (12) is bounded above with $\varepsilon K_3[1 + L(\theta(n))]$. Since it is assumed that $\mathbf{E}L(\theta(n)) < \infty$, it follows that $\mathbf{E}L(\theta(n+1)) < \infty$ and $\mathbf{E}\|H(n+1)\|^2 < \infty$. Thus by induction, it is proved for $n \le N_k$, $\mathbf{E}L(\theta(n)) < \infty$ and $\mathbf{E}\|H(n)\|^2 \le \infty$.

Since as $n \to \infty$, K_1 and K_2 go to infinity, instead of induction introduced above to prove the

bound of $\mathbf{E}\|H(n)\|^2$, another induction method is introduced. In the last iteration of previous induction, it is obtained that $\mathbf{E}L(\theta(N_k+1))<\infty$ and $\mathbf{E}\|H(N_k+1)\|^2<\infty$. Suppose that for some $n>N_k$, $\mathbf{E}L(\theta(n))<\infty$. Because the second term on the right side of (16) is F_n -measurable, we have $\mathbf{E}L(\theta(n+1))<\infty$. From (12) and (16), we have

$$\begin{aligned}&\varepsilon L'_\theta(\theta(n+1))\gamma(n+1)\\&\quad+\mathbf{E}_{n+1}(\varepsilon^2 H(n+1)^T\begin{bmatrix}1&0\\0&\Phi\end{bmatrix}H(n+1))\\&\le -k(\theta(n+1))\\&\quad+\mathbf{E}_n\left[b_3(n+1)\left\|Q^r(n+1)-\mathbf{E}_nQ^r(n+1)\right\|^2\right].\end{aligned}$$

Since the right side of inequality is $F(n+1)$ -measurable and converges to zero according to (16), the left side must be bounded. Consequently if $\gamma(n+1)<\infty$, $\mathbf{E}\|H(n+1)\|^2$ must be bounded, or $\mathbf{E}\|H(n+1)\|^2<\infty$. Therefore for $n>N_k$, it is also proved that $\mathbf{E}\|H(n)\|^2<\infty$. By these two inductions we have concluded $\mathbf{E}\|H(n)\|^2<\infty$ for each n.

3.1.3 Proof on the asymptotic rate of change conditions

Define

$$M^0(t)=\sum_{i=0}^{m(t)-1}\varepsilon(i)\delta M(i),\ \delta M(n)=H(n)-\mathbf{E}_nH(n),$$

where $m(t)$ denotes the unique value of n such that $t_n\le t<t_{n+1}$. Since it is proved that $\mathbf{E}\|H(n)\|^2<\infty$ for each n, it is obvious that $\delta M(n)$ must be bounded. Now we should prove that for each μ and T, the following equation hold.

$$\lim_{n\to\infty}P\left\{\sup_{j\ge n}\max_{0\le t\le T}\left|\sum_{i=m(jT)}^{m(jT+t)-1}\varepsilon(i)\delta M(i)\right|\ge\mu\right\}=0. \quad (18)$$

Assume that (18) does not hold, then we can choose a sequence $\{\delta M(n)\}$ such that

$$\lim_{n\to\infty}\max_{0\le t\le T}\left|\sum_{i=m(jT)}^{m(jT+t)-1}\varepsilon(i)\delta M(i)\right|\ge\mu.$$

Since

$$\begin{aligned}\max_{0\le t\le T}\left|\sum_{i=m(jT)}^{m(jT+t)-1}\varepsilon(i)\delta M(i)\right|&\le\sum_{i=m(jT)}^{m(jT+T)-1}\varepsilon(i)\left|\delta M(i)\right|\\&\le[m(jT+T)-m(jT)-1)\varepsilon(m(jT)]\delta M^m,\end{aligned}$$

where $\delta M^m=\max_{m(jT)\le i\le m(jT+1)-1}\left|\delta M(i)\right|$. It follows that

$$\lim_{n\to\infty}[m(jT+T)-m(jT)-1)\varepsilon(m(jT)]\delta M^m\ge\mu.$$

From Assumption 1) we know $\lim_{n\to\infty}\varepsilon(n)\to 0$. Then δM^m must be infinity and growing at least as $O(\varepsilon^{-1}(n))$. That contradicts the previous sentence that $\delta M(n)$ is bounded. Therefore (18) holds. According to Theorem 5.3.2 in [18], the following conditions for asymptotic rate of change hold:

$$\lim_{n\to 0}\sup_{j\ge n}\max_{0\le t\le T}\left|M^0(jT+t)-M^0(jT)\right|=0. \quad (19)$$

3.1.4 Completion of the proof

Applying definition of $\delta M(n)$ and $\gamma(n)$, we define the sequence of shifted processes as

$$\theta^n(t)=\theta(n)+\sum_{i=n}^{m(t_n+t)-1}\varepsilon\gamma(i)+\sum_{i=n}^{m(t_n+t)-1}\varepsilon\delta M(i). \quad (20)$$

We have proved that, (16) implies that $\theta(n)$ returns to the neighborhood of $\{\theta\,|\,k(\theta(t))=\Phi(1-\alpha)\mathbf{E}\left\|Q^r(t)-\mathbf{E}_nQ^r(t)\right\|^2\}$ infinitely often as $n\to\infty$. Since $Q^r(n)$ is bounded, and $\mathbf{E}_n\left\|Q^r(n)\right\|$ is nonincreasing over iterations, $\Phi(1-\alpha)\mathbf{E}\left\|Q^r(t)-\mathbf{E}_nQ^r(t)\right\|^2$ must be bounded. Define a set, $\Psi_\lambda=\{\theta:L(\theta)\le\lambda\}$, and let Ψ_λ^c represent the complement of the set Ψ_λ. Fix δ and $\Delta>2\delta+\Phi(1-\alpha)\mathbf{E}\left\|Q^r(n)-\mathbf{E}_nQ^r(n)\right\|^2>0$, and let τ denote a stopping time such that $\theta(\tau)\in\Psi_\delta$. Obviously for large $n>N_k$ which makes (16) hold, and $\varepsilon(n)$ become trivial, $\gamma(n)$ can not force $\theta(n)$ out of Ψ_Δ. Then the only way to force $\theta^n(t)$ out of Ψ_Δ infinitely often is by the effects of $\{\delta M_i, i>\tau\}$. But according to the definition of $M^0(t)$ and (19), it is implied that

$$\lim_{n\to 0}\sum_{i=\tau}^{m(n+t)}\varepsilon(i)\delta M(i)I_{\{\theta(i)\in\Psi_\Delta\}}=0. \tag{21}$$

Therefore for finite and large enough τ, these martingale difference terms can not force $\theta^n(t)$ from Ψ_δ to Ψ_Δ infinity often. This implies $\theta(n)$ must converge to the set $\{\theta \mid k(\theta(t))=\Phi(1-\alpha)\ \mathbf{E}\left\|Q^r(t)-\mathbf{E}_nQ^r(t)\right\|^2\}$ with probability one.

Remark 1: Since $\phi_1(n)$ and $\phi_2(n)$ are chosen as random variables satisfying continuous uniform distribution and nonzero expectation, $\mathbf{E}[b_3(n)\left\|Q^r(n)-E_nQ^r(n)\right\|^2]$ must be decreasing, if P^* is the global best solution. As n becomes large enough so that $b_3(n)\to\Phi(1-\alpha)^2$, the invariant set is $\{\theta \mid k(\theta(t))=\Phi(1-\alpha)^2\mathbf{E}\left\|Q^r(t)-\mathbf{E}_nQ^r(t)\right\|^2\}$.

Obviously the only way to make this set become $\{\theta \mid k(\theta(t))=0\}$ is to make $Q^r(n)\to 0$ as $n\to\infty$, or $P^d(n)$ and $P^g(n)$ approach P^*. The key factor to make $P_i^d(n)$ approach the global best solution P^* is to make particles explore the vicinity around P^*. Hence how to improve the opportunity of exploring around P^* is our motivation to introduce the random velocity $\xi(n)$ to PSO-CREV algorithm.

3.2. Properties caused by stochastic behavior

Based upon the proof presented in the previous section, two main properties of PSO-CREV resulting from stochastic components are proposed.

Property 1: Divergence ahead of Convergence

In the proof, it is mentioned that, before n is large enough to make (16) hold, the individual updating principle is nonconvergent so that particle will move away from the best position recorded by itself and its neighborhood. This phenomenon can be viewed as a strong exploration that all particles wander in the solution space and record the best solutions found so far. And when $n>N_k$, particles start to converge to the best solutions found so far.

Property 2: Additional Stochastic Search Velocity

The random velocity $\xi(n)$ is the key improvement on the PSO-CREV. A nonzero $\xi(n)$ is very useful to enhance **exploration** ability. Moreover the PSO-CREV with zero $\mathbf{E}\xi(n)$ converges faster than the one with nonzero $\mathbf{E}\xi(n)$. Hence in applications $\xi(n)$ with zero expectation is more preferable than nonzero one.

4. STUDY ON PARAMETERS AFFECTING PSO-CREV PERFORMANCE

In this section, we are going to decide the parameters by experimental methods. Firstly the designs of decreasing $\varepsilon(n)$ and dynamic stochastic search velocity $\xi(n)$ are proposed as follows.

1) Decreasing $\varepsilon(n)$

A broadly used form for deceasing ε satisfying Assumption 2) of Theorem 1 is expressed as.

$$\varepsilon(n)=\frac{a}{(1+n)^b}, \tag{22}$$

where a and b are two positive scalars.

2) Dynamic stochastic search velocity $\xi(n)$

A time-varying $\xi(n)=w(n)\bar{\xi}(n)$ is proposed, where $\bar{\xi}(n)$ represents a stochastic velocity with zero expectant and constant value range, $w(n)$ represents a time-varying positive coefficient, whose dynamic strategy can be designed freely. In this paper, the strategy of $w(n)$ is defined as

$$w(n)=\begin{cases}1, & n<\frac{3}{4}N_b,\\ \eta w(n-1), & n\ge\frac{3}{4}N_b,\end{cases} \tag{23}$$

where N_b represents the total number of iterations, and η is a positive constant less than 1. Hence when $n<\frac{3}{4}N_b$, a strong velocity is applied to the particles to enhance their exploration ability. And in the last quarter of iterations, the bound of $\xi(n)$ decreases iteration by iteration, so that $\xi(n)$ has trivial effect on the convergence of PSO-CREV finally.

Obviously all parameters referred in PSO-CREV are:

1) Constants in velocity updating: c_1 and c_2;
2) α used in position updating;
3) Parameters referred in (23): a and b;
4) The stochastic velocity with constant bound: $\bar{\xi}(n)$.

Since any values of these parameters satisfying requirements in Theorem 1 are valid, we have to use experimental methods to determine the optimal values of these parameters. Since the main improvement of PSO-CREV is to make PSO-CREV be more efficient to avoid local minima, the primary goal of optimizing parameters is to make PSO-CREV has strong exploration ability, while the convergence rate is

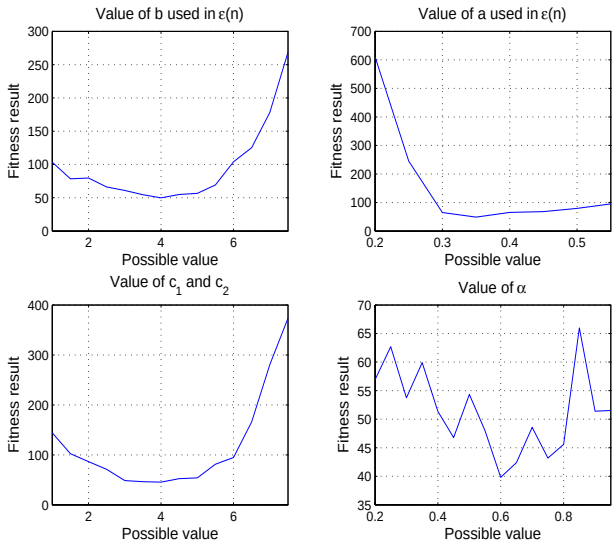

Fig. 3. Test results referred to all possible parameters.

concerned secondarily. Hence the fitness function with huge local minima around the best solution is chosen as

$$F(X)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right).$$

Because b determines the decreasing rate of $\varepsilon(n)$, the intension of stochastic search is limited by b. Hence the value of b should be determined firstly. In response to b, the second parameter to be optimized is a, which affects the span of duration of divergence ahead of convergence. Once the values of these two parameters are fixed, c_1 and c_2 are the next parameters needing optimization, because they play the roles of amplifying coefficients of cognitive and social components. Finally α is the last parameter for test in order to find the optimal value determining convergence behavior. In one word, the order of experiments for parameter optimization is shown as: $b \Rightarrow a \Rightarrow c_1, c_2 \Rightarrow \alpha$.

At the beginning, the initial parameters are set to be: $b=0.05$, $a=2$, $c_1=c_2=0.5$, $\alpha=0.95$. Then once a parameter is optimized, its value is set to the optimized one, and continue the next experiment for the next parameter. Each test will be repeated 25 runs independently. And there are 2000 iterations for each test. The results of all experiments are shown in Fig. 3 where in each subfigure, X-axis represents the possible values used for a test, Y-axis represents the fitness values with respect to different parameter values after 2000 iterations. It is concluded that the best parameter values are: $a=4$, $b=0.35$, $c_1=c_2=4$, $\alpha=0.6$.

5. BENCHMARKS ON PSO-CREV

To investigate the performance of PSO-CREV, in this section a series of benchmarks is proposed in which the optimization functions have different characters. Each experiment includes 2000 iterations. The dimension of solution space is set to $D=30$. All experiments are run 25 times. The results reported are the averages calculated from all 25 runs. For each benchmark, the sizes of swarm are chosen as 10, 20, and 30 respectively.

5.1. Test functions

Four functions are selected for testing, where o represents the shifted global optima, R represents an orthogonal matrix to rotate particle's coordinate. One can refer to [19] to find their values.

1) Shifted Griewank's function, $X \in [0 \quad 600]^D$.

$$F_1(X)=\sum_{i=1}^{D}\frac{z_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}})+1,\ (Z=X-o).$$

2) Shifted Schwefel's problem with noise in fitness, $X \in [-100 \quad 100]^D$.

$$F_2(X)=\left(\sum_{i=1}^{D}\left(\sum_{j=i}^{i}z_j\right)^2\right)\cdot(1+0.4\,|\,N(0,1)\,|),\ (Z=X-o),$$

where $N(0,1)$ represents a random value with Gaussian distribution.

3) Shifted Rosenbrock's function, $X \in [-50 \quad 50]^D$.

$$F_3(X)=\sum_{i=1}^{D}\left(100(z_i^2-z_{i+1})^2+(z_i-1)^2\right),$$
$$(Z=X+1_{D\times 1}).$$

4) Shifted rotated Rastrigin's function, $X \in [-5 \quad 5]^D$.

$$F_4(X)=\sum_{i=1}^{D}\left(Z_i^2-10\cos(2\pi z_i)+10\right),$$
$$(Z=(X-o)\cdot R).$$

5.2. PSO configurations

To compare PSO-CREV performance with other improvements of PSO, the following PSO algorithms are chosen for benchmarks, where the maximal velocity V_{max} is set to be the upper limit of the search range.

1) PSO-CREV ($\xi \neq 0$): The updating principle is of the form (7) with strategy of $\varepsilon(n)$ in the form of (22) and a decreasing $\xi(n)$ whose $w(n)$ is in the form of (23). All parameters are chosen as: $c_1=c_2=4$, $\alpha=0.6$, $a=4$, $b=0.35$, and

$\eta = 0.99$. The *lbest* version of neighborhood is used in which each particle exchanges information with four other particles [20].

2) PSO-CREV $(\xi = 0)$: All components are the same as PSO-CREV $(\xi \neq 0)$, except that $\xi = 0$.
3) GCPSO: The details of GCPSO can be found in [15]. And the parameters are chosen as: c_1=c_2=1.49, the inertia weight is 0.72. A dynamic strategy of f_c and s_c is adopted with f_c =5 and s_c =15 initially. The *gbest* version of neighborhood is used.
4) MPSO-TVAC: The details of MPSO-TVAC can be found in [14]. The ranges for c_1 and c_2 are 2.5→0.5 and 0.5→2.5 respectively. The inertia weight is set to change from 0.9 to 0.4 over the generations. The mutation step size is set to change from V_{max} to $0.1V_{max}$ over the search. The *lbest* PSO is used.
5) CPSO-H$_6$: The details of CPSO can also be found in [13]. Solution vector is split into six parts, where the inertia weight decreases from 0.9 to 0.4 over the search, c_1=c_2=1.49. The *lbest* PSO is employed.

5.3. Benchmark results

1) Shifted Griewank's Function ($F_1(X)$)
Griewank's function is multimodal. From the results of the tests shown in Table 1, the performance of PSO-CREV with nonzero ξ is very impressive for solving this kind of multimodal function.
2) Schwefel's problem with disturbance ($F_2(X)$)
Schwefel's problem is proposed to compare performance of all PSOs under the circumstance that there is disturbance in fitness function. Table 2 shows the results of the tests. Obviously when the stochastic noise is added to Schwefel's problem, the PSO-CREV $(\xi \neq 0)$ behaves better than other algorithms except CPSO-H$_6$.
3) Shifted Rosenbrock's function ($F_3(X)$) and shifted rotated Rastrigin's function ($F_4(X)$)
These two tests are proposed to check PSO-CREV performance for complex optimization functions, where Rosenbrock's function has a very narrow path from local minima to the global one, and in shifted rotated Rastrigin's function, huge local minima surround the global best solution, and the coordinates of particles are rotated. From the results shown in Table 3, PSO-CREV performs much better than other algorithms. Hence these two tests verify the improvement of PSO-CREV.

5.4. Summary on benchmarks

From the benchmarks proposed above, the following characters can be observed:

1) Relative to other PSO algorithms, the factor dominating the PSO-CREV performance is the additional velocity ξ, but not the swarm size.
2) Because of the first character, the performance of PSO-CREV seems much steady with respect to different swarm sizes.

Table 1. Comparison results of $F_1(X)$.

Griewank's function (F_1)			
Algorithm	M	Average	Std. Dev.
PSO-CREV ($\xi \neq 0$)	10	0.5045×10^{-2}	0.1343×10^{-2}
	20	0.2228×10^{-2}	0.7314×10^{-3}
	30	0.1166×10^{-2}	0.7028×10^{-3}
PSO-CREV ($\xi = 0$)	10	0.3043×10^{4}	31.7992
	20	0.2934×10^{4}	27.5756
	30	0.2905×10^{4}	19.6234
GCPSO	10	0.01681	0.3909×10^{-2}
	20	0.01623	0.3439×10^{-2}
	30	0.5123×10^{-2}	0.1413×10^{-9}
MPSO-TVAC	10	0.09561	0.01270
	20	0.04065	0.7432×10^{-2}
	30	0.02838	0.7187×10^{-2}
CPSO-H$_6$	10	0.02993	0.6107×10^{-2}
	20	0.04535	0.7549×10^{-2}
	30	0.02685	0.4624×10^{-2}

Table 2. Comparison results of $F_2(X)$.

Schwefel's problem (F_2)			
Algorithm	M	Average	Std. Dev.
PSO-CREV ($\xi \neq 0$)	10	0.1952×10^{3}	55.9361
	20	10.7958	1.0558
	30	4.9975	0.6864
PSO-CREV ($\xi = 0$)	10	0.5686×10^{6}	0.3285×10^{4}
	20	0.3613×10^{6}	0.1551×10^{4}
	30	0.2560×10^{6}	0.1265×10^{4}
GCPSO	10	0.4983×10^{6}	0.3496×10^{4}
	20	0.2372×10^{6}	0.1958×10^{4}
	30	0.1531×10^{6}	0.1202×10^{4}
MPSO-TVAC	10	0.2329×10^{6}	0.1493×10^{4}
	20	0.1349×10^{6}	0.1066×10^{4}
	30	0.9638×10^{5}	0.7587×10^{3}
CPSO-H$_6$	10	0.4011×10^{3}	73.8478
	20	0.02832	0.01122
	30	0.7171×10^{-7}	0.3962×10^{-7}

Table 3. Comparison results of $F_3(X)$ and $F_4(X)$.

		Shifted Rosenbrock's Function F_3		Shifted rotated Rastrigin's Function F_4	
Algorithm	M	Average	Std. Dev.	Average	Std. Dev.
PSO-CREV ($\xi \neq 0$)	10	40.2104	3.4169	76.2730	3.3208
	20	38.2621	4.7426	60.6241	3.0762
	30	35.2396	4.2082	49.8744	1.7776
PSO-CREV ($\xi = 0$)	10	0.9857×10^7	0.1506×10^7	0.3754×10^3	11.4447
	20	0.4335×10^6	0.1127×10^6	0.2821×10^3	11.5913
	30	0.2993×10^6	0.6292×10^5	0.2396×10^3	12.4231
GCPSO	10	63.5249	7.0700	0.2657×10^3	15.3048
	20	54.4399	8.6968	0.2178×10^3	12.3657
	30	52.7644	7.1727	0.1879×10^3	11.4271
MPSO-TVAC	10	78.3706	8.3019	0.1199×10^3	8.3585
	20	88.0143	23.2490	96.8357	5.7408
	30	57.5908	16.864	90.0168	7.8705
CPSO- H_6	10	55.2673	8.6518	0.1473×10^3	8.6037
	20	41.6887	5.5179	0.1725×10^3	12.2524
	30	21.3248	7.3055	0.1691×10^3	10.9591

6. APPLICATION OF PSO-CREV FOR RECURRENT NN TRAINING

In this section, an application about PSO-CREV in NN training is presented, where a kind of recurrent neural network is chosen as the optimization objective. The dimension of search space involved in PSO-NN training is always so large that PSO-NN training is viewed as a good platform to test performance of PSO algorithms. Roughly speaking, the main idea of PSO-NN training can be expressed as: Given a multi-layer neural network, all its weights are combined together to form a vector which is viewed as a solution vector in a solution space of PSO. Then a swarm is proposed whose particles represent such solution candidates. According to certain criterions, such as minimal root of mean square error (RMSE), all particles congregate to a position on which the coordinate represents the best solution they found. Then the best solution vectors are the best weights found by PSO.

6.1. Structure of Recurrent NN

A full connected recurrent neural network (FCRNN), as shown in Fig. 4, is trained to generate the following temporal trajectories,

$$y_1^d(t) = (0.35 - 0.005t)sin(0.5t) + 0.4,$$
$$y_2^d(t) = (0.25 + 0.005t)\cos(0.5t) + 0.4.$$

The discrete-time step is set to $\Delta t = 0.2,$ so that if the time ranges of the trajectories are limited within the interval (0,10], there are 50 steps within the interval. If there is no external input for FCRNN, the architecture of FCRNN is designed such that there are 20 nodes in the hidden layer and two output nodes in output layer respectively. Consequently there are 20×20 weights waiting for optimization. Moreover the NN training algorithm is also required to search the initial values of outputs of all nodes at the first step $t = 0.2.$ A sigmoid function is chosen as transfer function for all nodes. For the test, the root mean square error (RMSE) is chosen as function evaluation principle, which is of the form

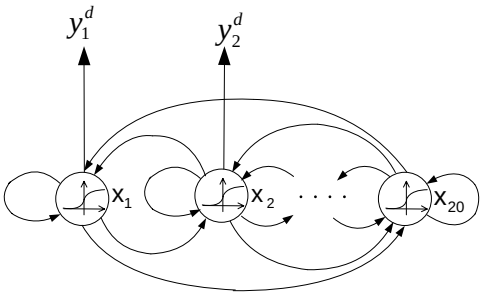

Fig. 4. Full connected recurrent neural network with 20 nodes.

$$RMSE = \left[\frac{1}{50}\sum_{k=1}^{50}\Big((y_1^d(k\Delta t) - x_1(k\Delta t))^2 + (y_2^d(k\Delta t) - x_2(k\Delta t))^2\Big)\right]^{1/2}. \quad (24)$$

6.2. Configurations of training algorithms

In addition to PSO-CREV ($\xi \neq 0$), other PSO algorithms referred in the benchmarks are all applied as comparison, except that PSO-CREV ($\xi(n) = 0$) is replaced by another evolutionary technique, genetic algorithm (GA), where the crossover probability is set to 0.8, and the mutation probability is chosen as 0.1. As a comparison with PSO-CREV, the number of chromosomes is chosen as 30, while the swarm size of all PSOs is set 10 and 20.

The configurations of all PSO algorithms are the same as those mentioned in the benchmarks. To test the main character of PSO-CREV that its particles have strong exploration ability to search solution space thoroughly in relative short iterations and with relative small population size, the number of iterations for NN training is limited within 1000 iterations, Accordingly η in (23) is set to 0.985 instead of 0.99, in order that $\xi(n)$ is suppressed quickly in the last quarter of iterations.

6.3. Results of FCRNN training

Since there are 420 values (including 400 weights and 20 initial values of all nodes) to be optimized, the dimension of solution space for PSOs and GA algorithms is 420. Each training algorithm is executed for 25 independent runs, so that the averages of the performance of all algorithms are shown in Table 4.

Obviously for standard GA, the number of 1000 iterations is too short to approach the global best solution. Hence the results of GA are very poor comparing with PSOs. The strong exploration ability brought by $\xi(n)$ makes PSO-CREV perform better than other algorithms except CPSO-H_6. But from Table 5 which shows the computation time of all algorithms relative to PSO-CREV, the computational cost of CPSO- H_6 is so huge that its computation time is more than twice of total time consumed by other algorithms. When swarm size is 10, the performance of PSO-CREV nearly catches up to CPSO- H_6. That means the exploration ability brought by ξ makes PSO-CREV perform as good as several swarms working together. Hence compared to CPSO which increases computational time greatly, PSO-CREV looks like a more "economical" way to enhance PSO performance.

Table 4. Comparison results of FCRNN training.

Algorithm	M	Average	Std. Dev.
PSO-CREV ($\xi \neq 0$)	10	0.04423	0.1137×10^{-2}
	20	0.04181	0.9124×10^{-3}
GCPSO	10	0.04841	0.1931×10^{-2}
	20	0.05104	0.1552×10^{-2}
MPSO-TVAC	10	0.07667	0.01924
	20	0.04421	0.2251×10^{-2}
CPSO-H_6	10	0.04305	0.9406×10^{-3}
	20	0.03232	0.6662×10^{-3}
GA	30	0.2914	0.1616×10^{-2}

Table 5. Comparison results of computation time.

Algorithm	Time
PSO-CREV ($\xi \neq 0$)	1
GA	1.0404
GCPSO	0.8530
MPSO-TVAC	0.8766
CPSO- H_6	8.0045

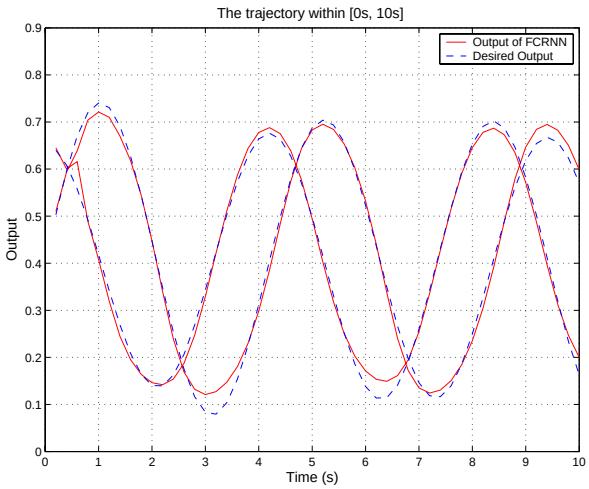

Fig. 5. The trajectory results from the FCRNN trained by PSO-CREV with 10 particles.

Finally as an example, the trajectory generated by the temporal sequence generator is shown in Fig. 5, which results from one run.

7. CONCLUSION

In this paper, an improved particle swarm optimization named PSO-CREV is proposed, which satisfies the regular stochastic recursion paradigm. Based on the proof of PSO-CREV convergence, two main characters are presented in terms of divergence ahead of convergence and controllable exploration ability. To select proper parameters used in practice, novel designs about $\xi(n)$ and $\varepsilon(n)$ are proposed. An experimental method is proposed to determine the values of parameters which guarantee strong exploration ability and good convergence rate. Through the benchmarks testing and the implementation on FCRNN training, it is verified that PSO-CREV can provide a competitive way to improve PSO algorithm.

REFERENCES

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. of 6th Int. Symp. on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. of IEEE Int. Conf. on Neural Network*, Perth, Australia, pp. 1942-1948, 1995.

[3] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 34, no. 2, pp. 997-1006, 2004.

[4] M. A. Abido, "Particle swarm optimization for multimachine power system stabilizer design," *Proc. of Power Engineering Society Summer Meeting*, pp. 1346-1351, 2001.

[5] L. Messerschmidt and A. P. Engelbrecht, "Learning to play games using a PSO-based competitive learning approach," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 280-288, 2004.

[6] Y. Li and X. Chen, "Mobile robot navigation using particle swarm optimization and adaptive NN," *Proc. of the First Int. Conf. on Natural Computation*, Changsha, China, LNCS 3612, pp. 554-559, 2005.

[7] H. M. Emara and H. A. Fattah, "Continuous swarm optimization technique with stability analysis," *Proc. of American Control Conference*, vol. 3, pp. 2811-2817, 2004.

[8] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[9] K. Yasuda, A. Ide, and N. Iwasaki, "Adaptive particle swarm optimization," *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 1554-1559, 2003.

[10] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317-325, 2003.

[11] B. Brandstäter and U. Baumgartner, "Particle swarm optimization-mass-spring system analogon," *IEEE Trans. on Magnetics*, vol. 38, no. 2, pp. 997-1000, 2002.

[12] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," *Proc. of IEEE Int. Conf. Evolutionary Computation*, Anchorage, AK, pp. 69-73, 1998.

[13] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239, 2004.

[14] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 240-254, 2004.

[15] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*, Ph.D. dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, 2002.

[16] B. Liu, L. Wang, Y. H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, no. 5, pp. 1261-1271, 2005.

[17] H. Robbins and S. Monro, "A stochastic approximation method," *Ann Math. Stat.*, vol. 22, pp. 400-407, 1951.

[18] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd Edition, Springer, 2003.

[19] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Technical Report*, Nanyang Technological University, Singapore, May 2005 and KanGAL Report #2005005, IIT Kanpur, India.

[20] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simple, maybe better," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3. pp. 204-210, 2004.

Xin Chen received the B.S. degree in Industrial Automation, and M.S. degree in Control Theory and Engineering from Central South University, Changsha, China in 1999 and 2002 respectively. He received the Ph.D. degree in Electromechanical Engineering from University of Macau, Macao SAR, China in 2007. He is now a Lecturer in Central South University. His research interests include swarm intelligence, multiple robot coordination.

Yangmin Li received the B.S. and M.S. degrees from the Mechanical Engineering Department, Jilin University, Changchun, China in 1985 and 1988 respectively. He received the Ph.D. degree from the Mechanical Engineering Department, Tianjin University, Tianjin, China in 1994. After that, he worked in South China University of Technology, International Institute for Software Technology of the United Nations University, University of Cincinnati, and Purdue University. He is currently a Professor in University of Macau. His research interests are mobile robots, swarm intelligence, and micro/nano manipulation. He is an IEEE Senior Member and a Member of ASME. He is a Council Member and Editor of Chinese Journal of Mechanical Engineering.