A Modified PSO Structure Resulting in High Exploration Ability With Convergence Guaranteed

Xin Chen and Yangmin Li, *Senior Member, IEEE*

***Abstract*—Particle swarm optimization (PSO) is a population-based stochastic recursion procedure, which simulates the social behavior of a swarm of ants or a school of fish. Based upon the general representation of individual particles, this paper introduces a decreasing coefficient to the updating principle, so that PSO can be viewed as a regular stochastic approximation algorithm. To improve exploration ability, a random velocity is added to the velocity updating in order to balance exploration behavior and convergence rate with respect to different optimization problems. To emphasize the role of this additional velocity, the modified PSO paradigm is named PSO with controllable random exploration velocity (PSO-CREV). Its convergence is proved using Lyapunov theory on stochastic process. From the proof, some properties brought by the stochastic components are obtained such as "divergence before convergence" and "controllable exploration." Finally, a series of benchmarks is proposed to verify the feasibility of PSO-CREV.**

***Index Terms*—Lyapunov theory, particle swarm optimization with controllable random exploration velocity (PSO-CREV), stochastic approximation, supermartingale convergence.**

I. Introduction

PARTICLE swarm optimization (PSO) is inspired from studies of various animal groups and has been proven to be a powerful competitor to other evolutionary algorithms such as genetic algorithms [1], [2]. PSO is widely used to solve nonlinear and multiobjective problems such as optimization of weights of neural networks (NN), electrical utility, computer games, and mobile robot path planning, etc. [3]–[7].

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. The particles evaluate their positions according to certain fitness functions at each iteration, and particles in a local neighborhood share memories of their "best" positions, then use those memories to adjust their own velocities and positions. Motivated by this, a second-order model is proposed to represent the traditional dynamics of particles. If we let M denote the size of the swarm, the current position of particle i is denoted by $X_i = [X_{i1}, X_{i2}, \ldots, X_{iD}]$, $i = 1, 2, \ldots, M$, where D is the dimension of the solution space, and its current velocity is denoted by v_i. Let $F(\cdot)$ denote the function for fitness estimation. Then, the updating principle is expressed as

$$v_{id}(n+1) = w_{id}v_{id}(n) + c_1 r_{1id}(n)\left[P^r_{id}(n) - X_{id}(n)\right] + c_2 r_{2id}(n)\left[P^g_{id}(n) - X_{id}(n)\right] \quad (1)$$

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

where $d = 1, \ldots, D$; $r_{1d} \sim U(0,1)$ and $r_{2d} \sim U(0,1)$ represent the two random numbers in the range of [0, 1]; c_1 and c_2 represent the acceleration coefficients; $P^r_i(n)$ represents the best position found by particle i so far; and $P^g_i(n)$ represents the global best position found by particle i's neighborhood.

The first part of (1) is the previous velocity, which provides the momentum for particles to pass through the search space, where w is the inertial weight. The second part is named as the "cognitive" component, which represents the personal thinking of each particle. The cognitive component of a particle takes the best position found so far by this particle as the desired input to make the particle move toward its own best positions. The third part is named as the "social" component, which represents the collaborative behavior of the particles to find the global optimal solution. The social component always pulls the particles toward the best position found by its neighbors.

In this paper, we will propose a modified PSO algorithm based on the traditional PSO structure, in which two major modifications are introduced.

On one hand, to make the PSO's algorithm satisfy the structure of the standard stochastic recursion algorithm, a decreasing proportional coefficient is introduced. Although this coefficient looks apparently like the definition of inertial weight, this decreasing coefficient goes to zero as the iteration increases, while the inertial weight is constant or dynamic, but greater than zero. We think that such a decreasing coefficient can be viewed as a reflection of emotion of "boring." Along with time running and congregation of swarm, particles believe that they are approaching the best solution; therefore, they would not like to take further exploration any more. We hope that the new algorithm is able to converge to an optimal solution as an iteration goes to infinity.

On the other hand, to balance a particle's exploration and exploitation, a random exploration velocity is introduced into the new PSO algorithm. Since this random velocity is relatively

Manuscript received May 10, 2006; revised January 17, 2007. This work was supported by the Research Committee of the University of Macau under Grant RG024/03-04S/LYM/FST. This paper was recommended by Associate Editor H. Ishibuchi.

The authors are with the Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macao S.A.R., China (e-mail: ya27407@umac.mo; ymli@umac.mo).

Digital Object Identifier 10.1109/TSMCB.2007.897922

independent of the cognitive and social components of the PSO updating principle, it can be designed at will. That means, one can adjust the PSO's exploration and exploitation ability to satisfy the requirements of different optimization tasks by employing different exploration velocities. Since this random velocity plays a very important role in this new PSO algorithm, the modified PSO algorithm is named PSO with controllable random exploration velocity (PSO-CREV).

The rest of this paper is organized as follows. In Section II, some significant previous works about PSO paradigms are introduced, such as the general representation of PSO and several PSO improvements, which are compared with the performance of PSO-CREV. In Section III, based on the general representation of PSO, we propose three modifications to illustrate the motivation of our design. According to the modifications, the description on PSO-CREV is introduced as a theorem in Section IV, whose convergence is proved using the theory of stochastic approximation [8], [9]. Based on the proof, several properties induced by stochastic components of PSO-CREV, particularly by the additional exploration ability, are summarized and tested through simulations. Experimental settings for the benchmarks and simulation strategies are introduced in Section V, and the results in comparison with other five improvements of PSO are presented in Section VI. Finally, conclusions are drawn in Section VII.

II. Preliminary Studies

A series of works has been done on the analysis and development of PSO since it was introduced in 1995. Most studies are related to two aspects.

The first aspect is mathematical analysis on the convergence of PSO. The original structure reflects an intuitive idea that a particle takes any position on which fitness value is better than where it currently is as the reference input is approached. Hence, from the view of a second-order system, a continuous-time model of PSO (CPSO) is proposed in which the updating principle is described as an ordinary differential equation [10]. However, since the original PSO is described as a discrete system, in which it is more convenient to use a discrete updating method in applications, most researchers prefer discrete system than continuous one [11]–[14]. Based on the two general representations [implicit representation (IR) and explicit representation], some important conclusions are obtained, such as the constriction coefficient which ensures that the reduced system be convergent [11].

Because these studies are based on a kind of reduced system that results from (1) and (2) by simplifying all stochastic factors, it is difficult to analyze the stochastic behavior of the PSO based on the reduced system, i.e., it is difficult to understand the behavior of exploration and exploitation brought by stochastic variables.

The second aspect of PSO research is to improve the PSO algorithm. Because the original model is similar to a kind of mobile multiagent system and each parameter describes a special character of natural swarm behavior, researchers can improve the performance of PSO according to the physical meanings of these parameters [15]–[19].

The first significant improvement is inertia weight [20], which results in fast convergence. Another significant improvement on PSO convergence is the constriction coefficient proposed in [11], which ensures the convergence of the reduced system as well as the PSO does. In fact, almost all improvements about PSO are based on these two basic improvements.

In conventional PSO algorithm, the search velocity $v(n)$ is always clamped within a range, which is denoted by $V_{\max}$. Given an optimization problem, the proper range of $V_{\max}$ for good performance is always limited and hard to be predicted. Hence, a PSO with decreasing $V_{\max}$ method (PSO-DVM) is developed [21], in which $V_{\max}$ is decreasing over time. By using this method, a large scale of searching is expected at the early steps, so that the population can remain in enough diversity profitable to converge to the global optimum. As the searching process continues, the searching scale is reduced to allow the solution to be found.

Another improved paradigm is called guaranteed convergence PSO (GCPSO) [22], which is developed specifically to address a drawback of the standard PSO. When the position of a particle equals its personal best position or the global best particle, the velocity is only influenced by the inertial term. Therefore, if this particle stays on the global best position, which is also the personal best position, for a number of iterations, its velocity tends to be zero and the particle stagnates. To solve this problem, the velocity update only for the global best particle changes, while the other particles remain as the original updating principle. In some degrees, it is viewed as a mutation method, where the mutation behavior is selective. That means only the particle in stagnation state is relocated randomly. A similar operation is adopted in the following PSO with a "mutation" and time-varying acceleration coefficient (MPSO-TVAC) paradigm.

MPSO-TVAC is developed by adopting two improvements [16]. The letter "M" in the name denotes mutation operation, which is used to make PSO escape from local best solution. But, different from GCPSO, in MPSO-TVAC, the particle for mutation is selected randomly from all particles when the global best solution is not improved at one iteration. Inspired by a time-varying inertia weight, another improvement in terms of TVAC is used to efficiently control the local search and convergence to the global optimum solution in addition to the time-varying inertia weight.

Different from both paradigms mentioned above in which one swarm is used for one optimization task, a cooperative PSO paradigm named as CPSO is developed to make several swarms work together to find the global best solution. The main idea of CPSO is using multiple swarms to optimize different components of the solution vector cooperatively. But, CPSO has such a drawback as stagnation caused by the restriction with only one swarm being updated at one time, i.e., only one subspace is searched at one time. The details of this drawback are described in [23]. To overcome this drawback, a PSO using a traditional algorithm is added, so that the CPSO algorithm is executed for one iteration, followed by one iteration of the PSO algorithm. This hybrid system is called CPSO-H$_K$, where K implies that the solution vector is split into K parts. Due to at one iteration, there are $K+1$ times of updating on swarms, and

comparing with other PSO paradigms, CPSO-H$_K$ needs more computational time while it brings a significant improvement in finding the global best solution.

These four improvements on PSO have the same character that, no matter how additional operations are employed, all of them use the basic updating principles with inertia weight or constriction coefficient to adjust particles' positions. But, in PSO-CREV, the original updating principle will be modified by introducing a decreasing coefficient that is different from the inertia weight and constriction coefficient. In addition, a relatively independent random velocity is employed to provide a strong exploration behavior while the convergence behavior is maintained.

III. Modifications on PSO Based on the Reduced System

A. Motivation of PSO-CREV

In the conventional PSO algorithm, two random variables r_{1d} and r_{2d} are employed to realize random exploration of particles. Rearranging terms of (1) and considering that $\mathbf{E}(r_1(n)c_1) = c_1/2$ and $\mathbf{E}(r_2(n)c_2) = c_2/2$, (1) can be expressed as

$$v_{id}(n+1) = w_{id}v_{id}(n) + \frac{c_1}{2}\left[P^r_{id}(n) - X_{id}(n)\right] + \frac{c_2}{2}\left[P^g_{id}(n) - X_{id}(n)\right] + v_{\mathrm{dis}}(n)$$

where

$$v_{\mathrm{dis}}(n) = \left(c_1 r_{1d}(n) - \frac{c_1}{2}\right)\left[P^r_{id}(n) - X_{id}(n)\right] + \left(c_2 r_{2d}(n) - \frac{c_2}{2}\right)\left[P^r_{id}(n) - X_{id}(n)\right].$$

Hence, the velocity updating consists of two parts. The first part is a stable component which makes a particle move to the target consisting of $P^r(n)$ and $P^g(n)$. The second part $v_{\mathrm{dis}}(n)$ looks like a disturbance with zero expected value. Moreover, as the particle approaches the target and $P^r(n) \to P^g(n)$, the values $[P^g(n) - X(n)]$ and $[P^r(n) - X(n)]$ go to zero. That means the disturbance is eliminated finally, and the particle converges to the best solution found so far. Obviously, it is $v_{\mathrm{dis}}(n)$ that enables particles to randomly search the solution space. According to the expression of $v_{\mathrm{dis}}(n)$, this random search has a drawback that its intensity is totally determined by the values of $P^r(n) - X(n)$ and $P^g(n) - X(n)$, which cannot be adjusted freely. Hence, there is a risk that if a swarm converges too quickly to a position, which may not be the global optima, particles also give up attempts for exploration.

Similar to $v_{\mathrm{dis}}(n)$, it seems reasonable to introduce another random velocity $\xi(n)$ into velocity updating, which is not dependent on $[P^g(n) - X(n)]$ and $[P^r(n) - X(n)]$, so that the random exploration behavior is enhanced and can be adjusted according to characters of different problems. Again, this velocity is viewed as a disturbance to the system, and it is easy to prove that since $\xi(n)$ is not dependent on the values of $[P^g(n) - X(n)]$ and $[P^r(n) - X(n)]$, the velocity updating will be not asymptotically convergent, unless it is prescribed that $\xi(n)$ is decreasing to zero. Hence, to employ $\xi(n)$, there need some mechanisms acting on the updating principle in order to guarantee convergence of updating principle. For this purpose, we develop some necessary modifications applied to the original PSO in order that the convergence of the system is maintained, even with $\xi(n)$ employed.

B. Modifications on the Level of Reduced System

Because PSO stems from the social behavior of mobile agents, whose model has been successfully described in forms of a reduced system [11], the modified PSO should satisfy the convergence of the reduced system too. Hence, the reduced system is selected as the start to introduce these modifications. According to IR in [11], a reduced system of a swarm is derived as

$$\begin{aligned} v_i(n+1) &= v_i(n) + \phi\left(P - X_i(n)\right) \\ X_i(n+1) &= X_i(n) + v_i(n+1) \end{aligned} \tag{3}$$

where P represents the best position found so far. In the following analysis, the subscript i is omitted for brevity. If a coefficient ε is introduced into the updating principle and let $Y(n) = X(n) - P$, the updating principle is changed to

$$\begin{aligned} v(n+1) &= \varepsilon v(n) - \varepsilon\phi Y(n) \\ Y(n+1) &= \varepsilon v(n) + (1 - \varepsilon\phi)Y(n). \end{aligned} \tag{4}$$

Let $\theta(n) = [v(n)\ Y(n)]^{\mathrm{T}}$ be the coordinate of particle and

$$A = \begin{bmatrix} \varepsilon & -\varepsilon\phi \\ \varepsilon & (1 - \varepsilon\phi) \end{bmatrix}$$

the control matrix of the system. In this case, we have $\theta(n+1) = A\theta(n)$. Thus, A determines the performance of the reduced system. The characteristic polynomial of matrix A is

$$Z^2 + (\varepsilon\phi - 1 - \varepsilon)Z + \varepsilon = 0 \tag{5}$$

and the eigenvalues of A are

$$\begin{aligned} e_1 &= \frac{1 + \varepsilon - \varepsilon\phi + \sqrt{(1 + \varepsilon - \varepsilon\phi)^2 - 4\varepsilon}}{2} \\ e_2 &= \frac{1 + \varepsilon - \varepsilon\phi - \sqrt{(1 + \varepsilon - \varepsilon\phi)^2 - 4\varepsilon}}{2}. \end{aligned} \tag{6}$$

Obviously, there are two parameters ϕ and ε which affect the eigenvalues.

As mentioned in Section III-A, if a random velocity $\xi(n)$ is employed to enhance the exploration behavior, there needs a mechanism to suppress $\xi(n)$ to guarantee convergence. Hence, the first modification is to make ε be a decreasing factor with increasing of n, and let it go to zero as $n \to \infty$. Apparently, from the view of function, $\varepsilon(n)$ looks like the definition of inertia weight, and from the view of the form, ε looks like

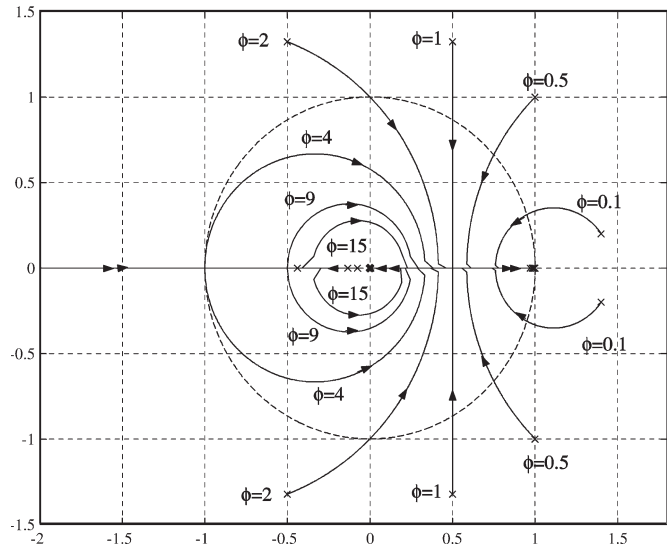

Fig. 1. Root locus plot with different ϕ as $\varepsilon(n) \to 0$.

the definition of the constriction coefficient. But, indeed, they are substantially different, because the inertia weight, even if a variant one, is always greater than zero, and the constriction coefficient is normally constant.

To illustrate the relationship between $\varepsilon(n)$ and the dynamics of the eigenvalue, an example of root locus of the reduced system is drawn in Fig. 1. From figure, two properties can be derived.

1) No matter what ϕ equals, as $\varepsilon(n)$ is approaching to 0, the roots go into the unit circle as complex roots.
2) No matter what ϕ equals, the roots finally converge to two real roots 0 and 1.

According to the stability theorem of discrete systems, which asserts that only the system with roots within the unit circle is convergent, along with decreasing $\varepsilon(n)$, one of two roots will converge to 1 so that the system maintains a sustained oscillation ultimately with frequency 1. That means the reduced system is nonconvergent. Therefore, to make the reduced system convergent, we propose the second modification through multiplying $Y(n)$ by a positive coefficient α that is less than 1, so that the reduced system is changed into

$$\begin{aligned} v(n+1) &= \varepsilon v(n) - \varepsilon\phi Y(n) \\ Y(n+1) &= \varepsilon v(n) + (\alpha - \varepsilon\phi) Y(n). \end{aligned} \tag{7}$$

Consequently, both roots converge to α and 0; hence, the reduced system is convergent. According to the discussions presented above, we draw the following conclusion.

Convergence of the reduced system referred to a PSO algorithm. If we generate a reduced system in terms of (7) from a certain PSO algorithm, the reduced system is stable, and X converges to the input P.

Based on the reduced system shown in (7), the modified PSO paradigm can be derived. To extend the reduced system to the updating principle, P involved in (3) is substituted by $(\phi_1 P^r + \phi_2 P^g)/(\phi_1 + \phi_2)$, 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, (7) is reexpressed as

$$\begin{aligned} v(n+1) =& \varepsilon(n) v(n) - \varepsilon(\phi_1 + \phi_2) \\ & \times \left(X(n) - \frac{\phi_1 P^r + \phi_2 P^g}{\phi_1 + \phi_2} \right) \\ =& \varepsilon(n) v(n) + c_1 r_1 \left(P^r - X(n) \right) \\ & + c_2 r_2 \left(P^g - X(n) \right) \\ X(n+1) - \frac{\phi_1 P^r + \phi_2 P^g}{\phi_1 + \phi_2} =& \varepsilon v(n) + (\alpha - \varepsilon\phi) \\ & \times \left(X(n) - \frac{\phi_1 P^r + \phi_2 P^g}{\phi_1 + \phi_2} \right) \\ =& \alpha X(n) + v(n+1) \\ & - \alpha \frac{\phi_1 P^r + \phi_2 P^g}{\phi_1 + \phi_2}. \end{aligned}$$

After arrangement of terms, the updating principle of $X(n+1)$ is expressed as

$$X(n+1) = \alpha X(n) + v(n+1) + \frac{1-\alpha}{\phi_1 + \phi_2} \left(\phi_1 P^r + \phi_2 P^g \right).$$

Obviously, comparing with the conventional PSO algorithm, there exists an additional term on the right hand of position updating, which is viewed as the third modification.

Until now, three modifications have been introduced based on the reduced system, which are summarized again as follows.

1) A decreasing $\varepsilon(n)$ is added into the velocity updating, which goes to zero as $n \to \infty$. It is designed to suppress random velocity $\xi(n)$, which is introduced in Section IV.
2) A positive coefficient α, which is less than 1, is added to the position updating in order to make the reduced system convergent.
3) An additional term $[(1-\alpha)/(\phi_1 + \phi_2)](\phi_1 P^r + \phi_2 P^g)$ is added to the expression of the position updating to complete the updating principle.

Although the reduced system implies that a PSO paradigm resulting from it is convergent, many properties about the dynamics of this PSO algorithm cannot be obtained because of the stochastic properties and, particularly, the effect of $\xi(n)$, which are all ignored in the reduced system. We need to use the theory of stochastic approximation to prove the convergence of the modified PSO paradigm, which is expressed as a theorem in the next section. From the proof, some useful properties brought by stochastic components are obtained in order to improve the PSO performance.

IV. PSO-CREV

A. Basic Description of PSO-CREV

Theorem 1: The 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 $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]^{\mathrm{T}}$,

where D represents the dimension of swarm space. The updating principle for each individual particle is defined as

$$\begin{aligned} v_{id}(n+1) &= \varepsilon_i(n)[v_{id}(n) + c_1 r_{1id}(n)(P_{id}^r(n) - X_{id}(n)) \\ &\quad + c_2 r_{2id}(n)\,(P_{id}^g(n) - X_{id}(n)) + \xi_{id}(n)] \\ X_{id}(n+1) &= \alpha X_{id}(n) + v_{id}(n+1) + \frac{1-\alpha}{\phi_{id}(n)} \\ &\quad \times (c_1 r_{1id}(n) P_{id}^r(n) + c_2 r_{2id}(n) P_{id}^g(n)) \end{aligned} \tag{8}$$

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

Suppose that 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 $r_{2id} \sim U(0,1)$. In addition, let $\Phi_{1i} = \mathbf{E}\phi_{1i}(n)$ and $\Phi_{2i} = \mathbf{E}\phi_{2i}(n)$, respectively.
5) $P_i^r(n)$ and $P_i^g(n)$ are always within a finite domain.

Then, the swarm must converge with probability one.

In addition, let $P^* = \inf_{\lambda \in \Sigma^D} F(\lambda)$ represent the global optimal positions in solution space Σ^D, and $\|P^*\| < \infty$. Then, the swarm will converge to P^* if $\lim_{n\to\infty} P_i^r(n) \to P^*$ and $\lim_{n\to\infty} P_i^g(n) \to P^*$.

B. Convergence Proof on PSO-CREV

The subscript i is omitted in the proof for clarity.

Since we are interested in the behavior that particles approach the global best solution, a new relative position is defined as $Y(n) = X(n) - P^*$. Substitute it into (8) and change (8) into

$$\begin{aligned} v(n+1) &= v(n) + [\,-(1-\varepsilon(n))\,v(n) - \varepsilon(n)\phi(n)Y(n) \\ &\quad + \varepsilon(n)\,(\phi_1(n)Q^r(n) + \phi_2(n)Q^g(n)) \\ &\quad + \varepsilon(n)\xi(n)] \\ Y(n+1) &= Y(n) + (\alpha - 1)Y(n) + v(n+1) \\ &\quad + \frac{1-\alpha}{\phi(n)}\,(\phi_1(n)Q^r(n) + \phi_2(n)Q^g(n)) \\ &= Y(n) + \Big[-(1-\alpha+\varepsilon(n)\phi(n))\,Y(n) \\ &\quad + \varepsilon(n)v(n) + \frac{1-\alpha+\varepsilon(n)\phi(n)}{\phi(n)} \\ &\quad \times (\phi_1(n)Q^r(n) + \phi_2(n)Q^g(n))\Big] \end{aligned} \tag{9}$$

where $Q^r(n) = P^r(n) - P^*$ and $Q^g(n) = P^g(n) - P^*$. Because of Assumption 5), $Q^r(n)$ and $Q^g(n)$ are finite.

Let $Q(n) = [1/\phi(n)][\phi_1(n)Q^r(n) + \phi_2(n)Q^g(n)]$ and $\theta(n) = [v^{\mathrm{T}}(n) \quad Z^{\mathrm{T}}(n)]^{\mathrm{T}} = [v^{\mathrm{T}}(n) \quad Y^{\mathrm{T}}(n) - \mathbf{E}_n Q^{\mathrm{T}}(n)]^{\mathrm{T}}$, where $\mathbf{E}_n$ denotes the expectation conditioned on the σ-algebra $\mathcal{F}_n$. Then, it follows that

$$\theta(n+1) = \theta(n) + \varepsilon(n)H(n) \tag{10}$$

where

$$H(n) = \frac{1}{\varepsilon(n)}\begin{bmatrix} h_1^{\mathrm{T}} & h_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

where

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

To make (10) a standard stochastic approximation algorithm, $\varepsilon(n)$ should decrease to zero over iterations. That is why we propose such a decreasing coefficient to the updating principle. A Lyapunov function is defined as

$$L(\theta(n)) = \frac{1}{2}\theta^{\mathrm{T}}\begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix}\theta = \frac{1}{2}\left(\|v\|^2(n) + \Phi\|Z\|^2(n)\right) \tag{11}$$

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

(1) Properties on the derivative of Lyapunov function.

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

$$\begin{aligned} \gamma(n) &= \mathbf{E}_n H(n) \\ &= \frac{1}{\varepsilon(n)}\begin{bmatrix} -(1-\varepsilon(n))\,v(n) - \varepsilon(n)\Phi Z(n) + \varepsilon(n)\Xi \\ -(1-\alpha+\varepsilon(n)\Phi)\,Z(n) + \varepsilon(n)v(n) \end{bmatrix}. \end{aligned} \tag{12}$$

Using a truncated Taylor series expansion, we have

$$\begin{aligned} \mathbf{E}_n L(\theta(n+1)) - L(\theta(n)) &= \varepsilon(n) L_\theta'^{\mathrm{T}}(\theta(n))\,\gamma(n) \\ &\quad + \mathbf{E}_n\left(\varepsilon^2(n) H(n)^{\mathrm{T}}\begin{bmatrix} 1 & 0 \\ 0 & \Phi \end{bmatrix} H(n)\right). \end{aligned} \tag{13}$$

Calculating the first term on the right side of (13) yields

$$\begin{aligned} &L_\theta'^{\mathrm{T}}(\theta(n))\gamma(n) \\ &\quad = \frac{1}{\varepsilon(n)}\mathbf{E}_n\Big\{-v^{\mathrm{T}}(n) \\ &\qquad \times [(1-\varepsilon(n))\,v(n) + \varepsilon(n)\Phi Z(n) - \varepsilon(n)\Xi] \\ &\qquad - \Phi Z^{\mathrm{T}}(n) \\ &\qquad \times [(1-\alpha+\varepsilon(n)\Phi)\,Z(n) - \varepsilon(n)v(n)]\Big\} \\ &\quad = -\frac{1}{\varepsilon(n)}\Big[(1-\varepsilon(n))\,\|v(n)\|^2 + \Phi(1-\alpha+\varepsilon(n)\Phi) \\ &\qquad \times \|Z(n)\|^2 - \varepsilon(n)v^{\mathrm{T}}(n)\Xi\Big]. \end{aligned} \tag{14}$$

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

$$\begin{aligned}\mathbf{E}\left[\left\|\tilde{H}(n)\right\|\left|\mathcal{F}(n)\right.\right]^2 = \frac{1}{\varepsilon^2(n)}\mathbf{E}_n\Big\{&a_1(n)\left\|v(n)\right\|^2\\ &+\ a_2(n)\left\|Z(n)\right\|^2 + a_3(n)\left\|Q(n)\right.\\ &-\left.\mathbf{E}_nQ(n)\right\|^2 + \varepsilon^2(n)\left\|\xi(n)\right\|^2\\ &+\ a_4(n)v^{\mathrm{T}}(n)Z(n) + \ a_5(n)\left[Q(n)\right.\\ &-\left.\mathbf{E}_nQ(n)\right] + a_6(n)\xi(n)\Big\}\end{aligned}\tag{15}$$

where

$$\begin{aligned}a_1(n) &= (1-\varepsilon(n))^2 + \Phi\varepsilon^2(n)\\ a_2(n) &= \Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2 + \left(\varepsilon(n)\phi(n)\right)^2\\ a_3(n) &= \Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2 + \left(\varepsilon(n)\phi(n)\right)^2\\ a_4(n) &= 2\varepsilon(n)\left[\phi(n)\left(1-\varepsilon(n)\right) - \Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right]\\ a_5(n) &= 2\left[\Phi\varepsilon(n)\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right.\\ &\quad\left.-\ \varepsilon(n)\phi(n)\left(1-\varepsilon(n)\right)\right]v^{\mathrm{T}}(n)\\ &\quad-2\left[\Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2+\varepsilon^2(n)\phi^2(n)\right]Z^{\mathrm{T}}(n)\\ &\quad-\varepsilon(n)\phi(n)\xi^{\mathrm{T}}(n)\\ a_6(n) &= -2\varepsilon(n)\left\{\left(1-\varepsilon(n)\right)v^{\mathrm{T}}(n)+\varepsilon(n)\phi(n)Z^{\mathrm{T}}(n)\right\}.\end{aligned}$$

Substituting (14) and (15) into (13), and using inequality $\|uv\| \le (1/2)(\|u\|^2+\|v\|^2)$, we obtain

$$\begin{aligned}\mathbf{E}_nL(\theta(n+1)) - L\left(\theta(n)\right) &\le b_1(n)\left\|v(n)\right\|^2 + b_2(n)\left\|Z(n)\right\|^2\\ &+\mathbf{E}_n\left[b_3(n)\left\|Q(n)-\mathbf{E}_nQ(n)\right\|^2\right] + \mathbf{E}_n\left[b_4(n)\left(Q(n)\right.\right.\\ &\left.\left.-\ \mathbf{E}_nQ(n)\right)\right] + b_5(n)\mathbf{E}_n\left\|\xi(n)\right\|^2 + \frac{1}{2}\sqrt{\varepsilon(n)}\|\Xi\|^2\end{aligned}\tag{16}$$

where

$$\begin{aligned}b_1(n) &= -\varepsilon(n) + (1+\Phi)\varepsilon^2(n) + \left(\frac{5}{2}+\varepsilon^2(n)-2\varepsilon(n)\right)\varepsilon^{\frac{3}{2}}(n)\\ b_2(n) &= -\Phi\alpha(1-\alpha) - \varepsilon(n)\Phi^2(2\alpha-1)\\ &\quad+\varepsilon^2(\Phi+2)\mathbf{E}_n\left(\phi^2(n)\right) + \sqrt{\varepsilon(n)}\mathbf{E}_n\\ &\quad\times\left[\phi(n)\left(1-\varepsilon(n)\right) - \ \Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right]^2\\ b_3(n) &= \Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2 + \left(\varepsilon(n)\phi(n)\right)^2\\ b_4(n) &= 2\left[\Phi\varepsilon(n)\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right.\\ &\quad\left.-\ \varepsilon(n)\phi(n)\left(1-\varepsilon(n)\right)\right]v^{\mathrm{T}}(n)\\ &\quad-2\left[\Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2+\varepsilon^2(n)\phi^2(n)\right]Z^{\mathrm{T}}(n)\\ &\quad-\varepsilon(n)\phi(n)\xi^{\mathrm{T}}(n)\\ b_5(n) &= \sqrt{\varepsilon(n)} + \varepsilon^2(n)\left(1+\mathbf{E}_n\left(\phi^2(n)\right)\right).\end{aligned}$$

If $\varepsilon(n)$ is large enough at the beginning, $b_1(n)$ and $b_2(n)$ must be positive so that $\mathbf{E}_nL(\theta(n+1)) - L(\theta(n)) \ge 0$. 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 $n\to\infty$, $\mathbf{E}_n[b_4(n)(Q(n)-\mathbf{E}_nQ(n))]\to 0$, and $b_5\mathbf{E}_n[\xi(n)]^2\to 0$. That means there exists $N_k<\infty$, such that when $n>N_k$, there is a positive nondecreasing function $k(\theta(n))$ to satisfy

$$\begin{aligned}&\mathbf{E}_nL\left(\theta(n+1)\right) - L\left(\theta(n)\right)\\ &\qquad\le -k\left(\theta(n)\right) + \mathbf{E}_n\left[b_3(n)\left\|Q(n)-E_nQ(n)\right\|^2\right].\end{aligned}\tag{17}$$

For a large n, (17) implies that the right side of (17) is negative outside of a neighborhood of the set $\{\theta|k(\theta(n)) \le \mathbf{E}_n[b_3(n)\|Q(t)-\mathbf{E}_nQ(n)\|^2]\}$. Outside such a decreasing neighborhood, $L(\theta(n))$ has the supermartingale property. Then, the supermartingale convergence theorem implies that a neighborhood of the set $\{\theta|k(\theta(n)) = \mathbf{E}_n[b_3(n)\|Q(n)-\mathbf{E}_nQ(n)\|^2]\}$ is recurrent, that is, $\theta(n)$ returns to it infinitely often with probability one. Since $\mathbf{E}_n[b_3(n)\|Q(n)-\mathbf{E}_nQ(n)\|^2]\to\Phi(1-\alpha)^2\mathbf{E}_n\|Q(n)-\mathbf{E}_nQ(n)\|^2$ as $n\to\infty$, $\theta(n)$ returns to neighborhood of $\{\theta|k(\theta(n)) = \Phi(1-\alpha)^2\mathbf{E}_n\|Q(n)-\mathbf{E}_nQ(n)\|^2\}$ infinitely often as $n\to\infty$.

(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 in [9, Lemma 5.4.1]. From (14) and (15), we observe that if $\varepsilon(n)<\infty$, $\mathbf{E}\|H(n)\|^2$ and $\mathbf{E}[L_\theta'^{\mathrm{T}}(\theta(n)\gamma(n))]$ are growing at most as $O(|\theta|^2)$. Therefore, there are two positive constants K_1 and K_2 such that

$$\mathbf{E}\|H(n)\|^2 + \mathbf{E}\left[L_\theta'^{\mathrm{T}}\left(\theta(n)\gamma(n)\right)\right] \le K_1L\left(\theta(n)\right) + K_2.\tag{18}$$

First, considering that $\mathbf{E}L(\theta(0))<\infty$ and (18), 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 (13), (18) implies that there is a real K_3 such that the right side of (13) is bounded above by $\varepsilon K_3[1+L(\theta(n))]$. 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 the induction introduced above to prove the bound of $\mathbf{E}\|H(n)\|^2$, another induction method is introduced. In the last iteration of the previous induction, it is obtained that $\mathbf{E}L(\theta(N_k+1))<\infty$ and $\mathbf{E}\|H(N_k+1)\|^2<\infty$. We suppose that for some $n>N_k$, $\mathbf{E}L(\theta(n))<\infty$. Because the second term on the right side of (17) is $\mathcal{F}(n)$-measurable, we have $\mathbf{E}L(\theta(n+1))<\infty$ from (17). From (13) and (17), we have

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

Since the right side of the inequality is $\mathcal{F}(n+1)$-measurable and converges to zero according to (17), 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) Proof on the asymptotic rate of changing conditions. Define

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

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 holds:

$$\lim_{n\to\infty} P\left\{\sup_{j\ge n}\max_{0\le t\le T}\left|\sum_{i=m(jT)}^{m(jT+t)-1}\varepsilon(i)\delta M(i)\right| \ge \mu\right\} = 0. \quad (19)$$

Assume that (19) 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

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

where $\delta M^m = \max_{m(jT)\le i\le m(jT+1)-1}|\delta M(i)|$, it follows that

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

From Assumption 2), 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 (19) holds. According to the study in [9, Th. 5.3.2], the following conditions for asymptotic rate of change hold:

$$\lim_{n\to\infty}\sup_{j\ge n}\max_{0\le t\le T}\left|M^0(jT+t) - M^0(jT)\right| = 0. \quad (20)$$

(4) Completion of the proof.

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

$$\theta^n(t) = \theta(n) + \sum_{i=n}^{m(t_n+t)-1}\varepsilon\gamma(i) + \sum_{i=n}^{m(t_n+t)-1}\varepsilon\delta M(i). \quad (21)$$

We have proved that (17) implies that $\theta(n)$ returns to the neighborhood of $\{\theta|k(\theta(n)) = b_3(n)\mathbf{E}\|Q(n) - \mathbf{E}_n Q(n)\|^2\}$ infinitely often as $n \to \infty$. Define a set as $\Psi_\lambda = \{\theta : L(\theta) \le \lambda\}$, and let Ψ_λ^c represent the complement of the set Ψ_λ. Fix δ and $\Delta > 2\delta + \Phi(1-\alpha)^2\mathbf{E}\|Q(n) - \mathbf{E}_n Q(n)\|^2 > 0$, and let τ denote a stopping time such that $\theta(\tau) \in \Psi_\delta$. Obviously, for large $n > N_k$ which makes (17) hold, and $\varepsilon(n)$ become trivial, $\gamma(n)$ cannot 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 (20), it is implied that

$$\lim_{n\to\infty}\sum_{i=\tau}^{m(n+t)}\varepsilon(i)\delta M(i)I_{\{\theta(i)\in\Psi_\Delta\}} = 0. \quad (22)$$

Therefore, for finite and large enough τ, these martingale difference terms cannot force $\theta^n(t)$ from Ψ_δ to Ψ_Δ infinity often. This implies that $\theta(n)$ must converge to the set $\{\theta|k(\theta(n)) = b_3(n)\mathbf{E}\|Q(n) - \mathbf{E}_n Q(n)\|^2\}$, or finally to $\{\theta|k(\theta(n)) = \Phi(1-\alpha)^2\mathbf{E}\|Q(n) - \mathbf{E}_n Q(n)\|^2\}$, with probability 1.

Remark 1: Although it is proved that the updating principle will converge with probability 1, it seems that particle will converge to a region but not a position. Moreover, from the definition $\theta(n) = [v(n)\ Y(n) - \mathbf{E}_n Q(n)]^{\mathrm{T}} = [v(n)\ X(n) - P^* - \mathbf{E}_n Q(n)]^{\mathrm{T}}$, if we do not know what the value of $Q(n)$ is, apparently the particle converges to an arbitrary position in solution space. This delusion results from the definition of $Y = X - P^*$, where P^* is defined as the global optimum which is unknown. Now, we analyze where a particle actually converges to. Instead of the original definition, P^* is redefined as a combined best record found so far, i.e., $P^* = (\Phi_1 P^r + \Phi_2 P^g)/(\Phi_1 + \Phi_2)$. It follows that $Q(n) = [(p^r(n) - p^g(n))/(\Phi_1 + \Phi_2)] \cdot [(\Phi_2\phi_1 - \Phi_1\phi_2)/(\phi_1 + \phi_2)]$. Then, its expected value and variance are computed as

$$\begin{aligned}\mathbf{E}\left(Q(n)\right) &= \frac{1}{2}\left(P^r(n) - P^g(n)\right) \\ &\quad \times \left\{\frac{1}{\Phi_1\Phi_2}\left[\left(\Phi_1^2 - \Phi_2^2\right)\ln(2\Phi_1 + 2\Phi_2)\right.\right. \\ &\quad \left.\left. - \left(\Phi_1^2\ln(2\Phi_1) - \Phi_2^2\ln(2\Phi_2)\right)\right] - \frac{\Phi_1 - \Phi_2}{\Phi_1 + \Phi_2}\right\} \\ \mathbf{E}\left[Q(n) - \mathbf{E}\left(Q(n)\right)\right]^2 &= \mathbf{E}\left(Q^2(n)\right) - \left[\mathbf{E}\left(Q(n)\right)\right]^2 \\ &= \frac{1}{\Phi_1\Phi_2(\Phi_1 + \Phi_2)^2} \\ &\quad \times \left(P^r(n) - P^g(n)\right)^2 \\ &\quad \times \left[\left(\Phi_2^3\Phi_1 + \Phi_1^3\Phi_2 + \Phi_1^2\Phi_2^2\right)\right. \\ &\quad - \left(\Phi_1 + \Phi_2\right)\left(\Phi_1^3 + \Phi_2^3\right) \\ &\quad \times \ln(2\Phi_1 + 2\Phi_2) \\ &\quad + \Phi_2^3(\Phi_1 + \Phi_2)\ln(2\Phi_2) \\ &\quad \left. + \Phi_1^3(\Phi_1 + \Phi_2)\ln(2\Phi_1)\right] \\ &\quad - \left[E\left(Q(n)\right)\right]^2.\end{aligned}$$

Considering the definition of Y, we can finally conclude that particle must uniformly converge to the vicinity around $[(\Phi_1 P^r(n) + \Phi_2 P^g(n))/(\Phi_1 + \Phi_2)] + E(Q(n))$ with the width satisfying the condition $\{\theta(n)|k(\theta(n)) \le b_3(n)\mathbf{E}_n[Q(n) - E(Q(n))]^2\}$.

In addition, from the expressions of expected value and variance, it is observed that to make the width $b_3(n)E[Q(n) - E(Q(n))]^2$ equal to zero, i.e., particle is uniformly asymptotically stable, the particle should fulfill at least one of the

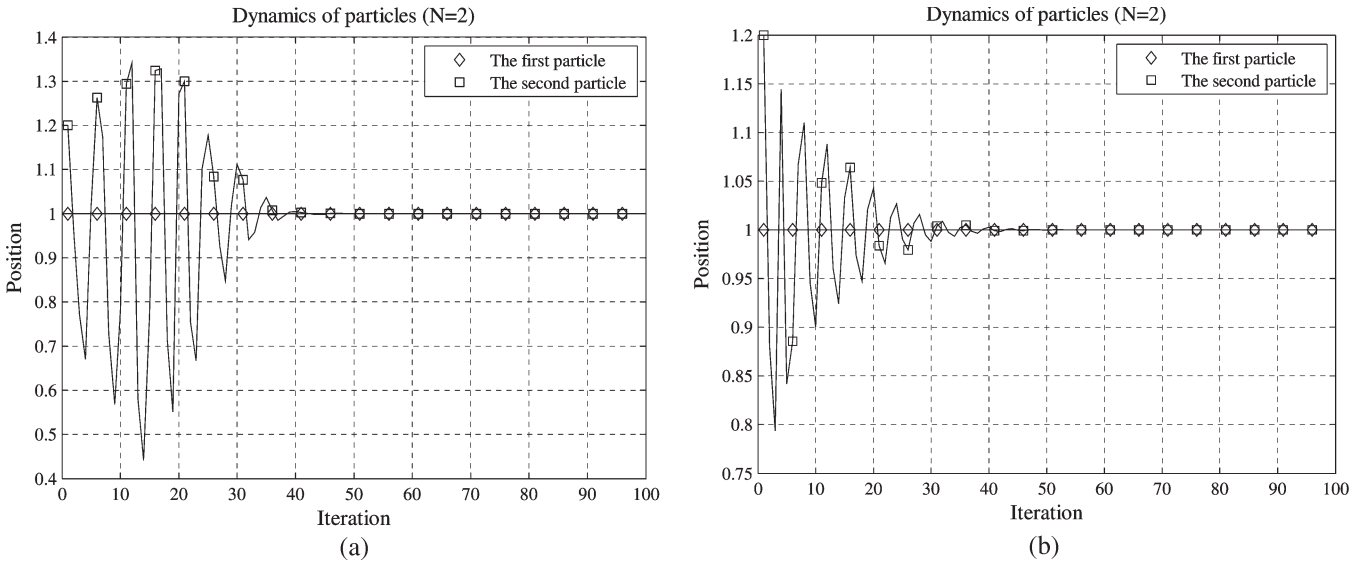

Fig. 2. Dynamics of a swarm with two particles and its reduced system. (a) Dynamics of particle 2 in the real PSO-CREV. (b) Dynamics of particle 2 in the reduced system.

following conditions: 1) $P^r(n) \to P^g(n)$, which means the best record found by the particle is also the best solution found by its neighborhood; 2) $\Phi_1 = 0$ or $\Phi_2 = 0$, which means setting $\phi_1(n)$ or (not "and") $\phi_2(n)$ to zero makes the particle asymptotically converge.

Remark 2: Now, come back to the original definition of P^*, which is defined as the global best solution $P^* = \inf_{\lambda \in \Sigma^D} F(\lambda)$, in order to analyze how to make particle converge to the global best solution. As mentioned above, because in the updating principle there is no explicit information about the global best solution P^*, the proof *does not* assert that $\lim_{n\to\infty} Q(n) = 0$ (here $Q(n) = [1/\phi(n)[\phi_1(n)Q^r(n) + \phi_2(n)Q^g(n)]$). When n becomes large enough such that $b_3(n) \to \Phi(1-\alpha)^2$, the invariant set is expressed as $\{\theta | k(\theta(n)) = \Phi(1-\alpha)^2 \mathbf{E}\|Q(n) - \mathbf{E}_n Q(n)\|^2\}$. Obviously, the only way to make this set become $\{\theta | k(\theta(n)) = 0\}$ is to make $Q(n) \to 0$ as $n \to \infty$, or $P^r(n)$ and $P^g(n)$ approach P^*. Because $P_i^g(n)$ represents the result of exchanging $P_i^r(n)$ among particles, the key factor to make $P_i^r(n)$ approach the global best solution P^* is to make the particles' trajectories close to P^*, or to make the 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 the PSO algorithm.

C. Properties and Discussions About PSO-CREV

Based upon the proof, two main properties of PSO-CREV resulting from stochastic components are introduced first. Then, the factors affecting the performance of PSO-CREV are discussed. Based on them, the algorithm is introduced.

1) Two Important Properties:

Property 1—Divergence Before Convergence: In the proof, it is mentioned that before n is large enough to make (17) hold, the individual updating principle is nonconvergent so that the particle will move away from the best position recorded by itself and its neighborhood. But, during this divergent process, particles are still recording their individual best solutions and exchanging information with each other. Hence, this phenomenon can be viewed as an exploration that all particles wander in the solution space and record the best solutions found so far. Also, when $n > N_k$, particles start to converge to the best solutions found so far.

To clearly illustrate this property of "divergence before convergence," a simple test is proposed, where a simple swarm, including two particles, is designed to get the best solution of a fitness function

$$F(X) = (1 - X)^2.$$

Obviously, the global best solution is $P^* = 1$. The parameters used in this simulation are chosen as $c_1 = c_2 = 1.75$, $\alpha = 0.95$, and $\varepsilon(n) = 1/(n+1)^{0.15}$. To avoid the effect of $\xi(n)$, let $\xi(n) = 0$. The first particle is located initially on the global best position where $X_1 = 1$, while the second one is not on the global best position. Obviously, particle 1 will be static permanently. In addition, particle 2 must approach the first particle due to interaction between them. Fig. 2(a) shows its real trajectory. As a comparison, the trajectory of the reduced system with respect to particle 2 is drawn in Fig. 2(b).

If $\varepsilon(1) \leq 1$ and $\phi < 4$, the dynamics of the reduced system is a strictly damped oscillation just as Fig. 2(b) shows. But, in Fig. 2(a), it is observed that particle 2 diverges from the global best solution founded by particle 1 at the beginning. In addition, after certain iterations, about 12 iterations, it starts to converge to the best position.

In a sense, this temporary divergence property can be viewed as an inherent exploration ability. But, this behavior is uncontrollable, because it is hard to compute the duration of a divergent interval to design $\varepsilon(n)$. Also, without $\xi(n)$, the intensity of stochastic behavior is still dependent on cognitive and social components, so that the basic drawback in terms of the uncontrollable convergence rate is not improved by this property.

Property 2—Controllable Exploration and Convergence: $\xi(n)$ in PSO-CREV is a stochastic component which can be designed freely. Obviously, without the additional stochastic behavior, i.e., $\xi(n)=0$, the PSO-CREV behaves much like the traditional PSO with relatively fast convergent speed, so that the intensity of exploration behavior is weakened quickly. Just as mentioned in Remark 2, because the global optima is unknown, the way to make particles converge to P^* is improving opportunities to explore the vicinity of P^*. Hence, a nonzero $\xi(n)$ is very useful to enhance exploration ability. At the same time, the proof of Theorem 1 reveals that this stochastic velocity will not change the convergence of PSO-CREV. Moreover, let us consider the condition that $\mathbf{E}\xi(n)=0$. Substituting it into the updating principle (14) yields

$$\mathbf{E}\left(L_\theta^{\mathrm{T}}(\theta(n)\gamma(n)\right)=-\frac{1}{\varepsilon(n)}\Big[(1-\varepsilon(n))\left\|v(n)\right\|^2 + \Phi\left(1-\alpha+\varepsilon(n)\Phi\right)\left\|Z(n)\right\|^2\Big]. \quad (23)$$

Therefore, (16) becomes

$$\begin{aligned}&\mathbf{E}_n L\left(\theta(n+1)\right)-L\left(\theta(n)\right)\\ &\le b_1(n)\left\|v(n)\right\|^2+b_2(n)\left\|Z(n)\right\|^2\\ &\quad+\mathbf{E}_n\left[b_3(n)\left(Q(n)-\mathbf{E}_nQ(n)\right)^2\right]\\ &\quad+\mathbf{E}_n\left[b_4(n)\left(Q(n)-\mathbf{E}_nQ(n)\right)\right]+b_5(n)\mathbf{E}_n\left\|\xi(n)\right\|^2 \quad (24)\end{aligned}$$

where

$$\begin{aligned}b_1(n)=&\ -\varepsilon(n)+(1+\Phi)\varepsilon^2(n)+\varepsilon^{\frac{3}{2}}(n)\\ b_2(n)=&\ -\Phi\alpha(1-\alpha)-\varepsilon(n)\Phi^2(2\alpha-1)\\ &+\varepsilon^2(\Phi+1)\mathbf{E}_n\left(\phi^2(n)\right)\\ &+\sqrt{\varepsilon(n)}\mathbf{E}_n\left[\phi(n)\left(1-\varepsilon(n)\right)\right.\\ &\qquad\left.-\Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right]^2\\ b_3(n)=&\,\Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2+\left(\varepsilon(n)\phi(n)\right)^2\\ b_4(n)=&\,2\left[\Phi\varepsilon(n)\left(1-\alpha+\varepsilon(n)\phi(n)\right)\right.\\ &\qquad\left.-\varepsilon(n)\phi(n)\left(1-\varepsilon(n)\right)\right]v^{\mathrm{T}}(n)\\ &-2\left[\Phi\left(1-\alpha+\varepsilon(n)\phi(n)\right)^2+\varepsilon^2(n)\phi^2(n)\right]Z^{\mathrm{T}}(n)\\ &-\varepsilon(n)\phi(n)\xi^{\mathrm{T}}(n)\\ b_5(n)=&\,\varepsilon^2(n).\end{aligned}$$

The rest of the proof is the same as mentioned in Section IV-B. Obviously, if other conditions are the same, $b_1(n)$ and $b_2(n)$ in PSO-CREV with zero $\mathbf{E}\xi(n)$ decrease faster than those in the PSO-CREV with nonzero $\mathbf{E}\xi(n)$, so that 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 a nonzero one. It should be mentioned that even if $\mathbf{E}\xi(n)\neq 0$, this nonzero expectation $\xi(n)$ does not change the convergence of PSO-CREV, but makes particles behave themselves with a directional exploration.

In Theorem 1, the only requirement about $\xi(n)$ is that it is bounded. But, the bound of $\xi(n)$ can be time-varying. Because of the assumption that $\sum_{n=0}^{\infty}\varepsilon(n)=\infty$, with the increasing of n, the decreasing gradient of $\varepsilon(n)$ becomes smaller and smaller. That means, if the bound of $\xi(n)$ is constant, as n increases, $\varepsilon(n)\xi(n)$ may be kept strong enough to overwhelm the convergence behavior brought by the cognitive and social components, so that the convergence of PSO-CREV will be delayed significantly. To overcome this drawback, a time-varying $\xi(n)=w(n)\bar{\xi}$ is proposed instead of the constant one, where $\bar{\xi}$ represents a stochastic velocity with zero expectation and constant value range; $w(n)$ represents a time-varying positive coefficient, whose dynamic strategy can be designed freely. For example, the following strategy of $w(n)$ looks very reasonable to balance exploration and convergence:

$$w(n)=\begin{cases}1, & n<\frac{3}{5}N_b\\ \eta w(n-1), & n\ge\frac{3}{5}N_b\end{cases} \quad (25)$$

where N_b represents the total number of iterations and η is a positive constant less than 1, which makes $\xi(n)$ be a decreasing random velocity when $n\ge(3/5)N_b$. Hence, when $n<(3/5)N_b$, a strong velocity is applied to the particles to enhance their exploration ability. Also, in the last 40% of iterations, the bound of $\xi(n)$ decreases iteration by iteration, so that $\xi(n)$ has a trivial effect on the convergence of PSO-CREV finally. In addition, this weakened velocity benefits particles to exploit the vicinity around the best solution found so far.

Such a random velocity $\xi(n)$ is the key modification on the PSO-CREV. A properly designed $\xi(n)$ will balance the convergence rate and exploration ability of PSO-CREV.

2) Factors of Affecting the Performance of PSO-CREV:

Dynamic strategy of $\varepsilon(n)$*:* Just as mentioned in Theorem 1, Assumption 2) presents that $\varepsilon(n)\to 0$ with n increasing, and $\sum_{n=1}^{\infty}\varepsilon(n)=\infty$. According to this requirement, a popular form for $\varepsilon(n)$ is expressed as

$$\varepsilon(n)=\frac{a}{(n+1)^b} \quad (26)$$

where a and b are positive scalars which determine the convergence rate of PSO-CREV, where $b<1$. Generally, the larger the value of a is, the stronger the divergence behavior of PSO-CREV shows at the beginning. That means a large a makes particle disperse widely and delays the threshold of convergence N_k. At the same time, b plays a very important role for determining the convergence speed. A small b makes PSO-CREV converge slowly, because $\varepsilon(n)$ decreases slowly. Hence, there exists a dilemma in choosing b. Choosing a small b makes PSO-CREV with strong exploration ability but weak convergent speed, while a large b makes PSO-CREV with the reverse character. To balance exploration and convergence speed, the value of b should be designed according to the strategy of $w(n)$, in order that PSO-CREV shows a relative active exploration at the beginning and a relative fast convergence during the last iterations.

To illustrate PSO-CREV's performance under different b, we propose a simple simulation where the number of iterations is

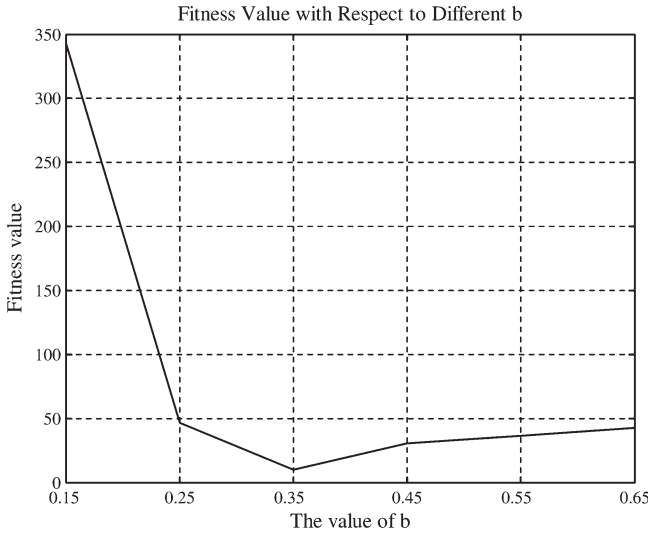

Fig. 3. Comparison about the performance of PSO-CREV with respect to different b.

fixed, and the strategy of $w(n)$ is chosen as (25), where $\eta = 0.993$. The dynamics of $\varepsilon(n)$ is of the form (26), where $a = 3.5$. The fitness function is chosen as

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

There are many local minima around the global best solution $P^* = 0$. Six values of b are selected to compare their performance. Each test is repeated 50 runs. In addition, the average fitness values of 50 runs with respect to different b are shown in Fig. 3. Other parameters used in the simulation are $c_1 = c_2 = 3.5$ and $\alpha = 0.95$. The number of iteration is 2000, and the dimension of the solution space is 10.

From the figure, we observe that if the number of iterations is 2000 and the strategy of $w(n)$ is of the form (25), the PSO-CREV with b around 0.35 performs better than other PSO-CREVs.

Clamped velocity: An upper limit of velocity of a particle, or clamped velocity $v_{\max}$, can be acted on the updating principle to speed up the convergence. By using $v_{\max}$, a new velocity $\tilde{v}(n+1)$ is defined as

$$\tilde{v}(n+1) = k(n)v(n+1)$$

where $k(n)$ is a proportional coefficient such that $k(n) = 1$, if $\|v_d(n+1)\| \le v_{\max}$, $d = 1, \ldots, D$, and $k(n) = v_{\max}/\|v_d(n+1)\|$, if $\|v_d(n+1)\| > v_{\max}$. Obviously, $k(n) \le 1$. Substituting $\tilde{v}(n+1)$ into (8) and letting $\tilde{\varepsilon}(n) = k(n)\varepsilon(n)$, we have

$$\begin{aligned} v_i(n+1) =& \tilde{\varepsilon}(n)\left[v_i(n) + c_{i1}r_{i1}(n)\left(P_i^r(n) - X_i(n)\right)\right. \\ &\left. + c_{i2}r_{i2}(n)\left(P_i^g(n) - X_i(n)\right) + \xi_i(n)\right] \\ X_i(n+1) =& \alpha X_i(n) + v_i(n+1) + \frac{1-\alpha}{\phi_i(n)} \\ &\times \left(c_{i1}r_{i1}(n)P_i^r(n) + c_{i2}r_{i2}(n)P_i^g(n)\right). \end{aligned} \quad (27)$$

The only difference between (8) and (27) is that $\varepsilon(n)$ is replaced by $\tilde{\varepsilon}(n)$. Since $k(n) \le 1$, it holds that $\tilde{\varepsilon}(n) \le \varepsilon(n)$, and $\tilde{\varepsilon}(n) \to 0$, as $n \to \infty$. Therefore, PSO-CREV with clamped velocity also converges with probability one. In fact, since $\tilde{\varepsilon}(n) \le \varepsilon(n)$, $b_1(n)$ and $b_2(n)$ decrease much quickly, so that the velocity-cramped PSO-CREV converges faster than the original PSO-CREV. But, the bounded velocity also induces lower exploration ability, so that particles may not be able to find the global best solution. The following simulation shows this phenomenon. A fitness function is of the form

$$F(X) = (X - 60)(X - 25)(X - 10)(X + 20)$$

which has two minima, 48.1190 and -9.4542, where the first one is the global optimal solution. Two kinds of PSO-CREV are designed, in which each swarm includes five particles. For the first swarm, a clamped velocity $v_{\max} = 35$ is employed, while the other swarms have no limit on their velocities. Except the clamped velocity, all parameters used in both PSO-CREVs are arranged as follows. $\varepsilon(n)$ is in the form of $\varepsilon(n) = 3.5/(n+1)^{0.4}$. $c_1 = c_2 = 2$, $\alpha = 0.95$. In addition, all particles are distributed within $[-10, 0]$ initially. For each PSO-CREV, the test is repeated 50 times to get numbers of successful runs and failing runs, respectively. To eliminate the effect of random exploration velocity, $\xi(n)$ in both PSO-CREVs is set to zero. The number of iterations for each run is set to 120, so that there is clear difference in convergent rate between them. Table I shows the test results. It is observed that, because it is only in 30 times out of 50 times that the first swarm can find the global best solution 48.1190, while the second one succeeds in all 50 runs, the PSO-CREV with clamped velocity performs much worse than the second one. Therefore, the clamped velocity does decrease the opportunities that particles approach the global solution. But, if only considering successful runs, because the average minimal fitness value and the standard deviation found by the PSO-CREV with clamped velocity are less than those found by another PSO-CREV, it is also concluded that the clamped velocity is able to improve the convergent speed of PSO-CREV. Therefore, a proper design of $v_{\max}$ will balance the exploration and convergent rate.

Completed description of PSO-CREV algorithm: Based on the analysis mentioned above, the PSO-CREV algorithm is presented by the pseudocode shown in Fig. 4.

V. Experimental Setup

In order to compare the different algorithms of PSO, in this section, we propose a series of benchmarks, so that PSO-CREV's performance is tested under several situations in terms of huge local minimums, stochastic disturbance, rotated configuration, and hybrid composition function.

Because $\xi(n)$ provides a strong exploration ability, it seems that PSO-CREV needs less iterations to search the best solution than the traditional PSO. Hence, here, we only let each experiment run for 2000 iterations so that we can observe the superiority of PSO-CREV clearly. The dimension of the solution space is set to 30. Moreover, to test the performance in high-dimensional optimization, another dimension 100 is

TABLE I
CLAMPED VELOCITY COMPARISON

	PSO-CREV with clamped velocity 50	PSO-CREV without clamped velocity
Total runs	50	50
Successful runs	30	**50**
Average fitness	$\mathbf{-7.132336 \times 10^5}$	-7.132320×10^5
Std. Dev. of fitness	$\mathbf{0.3610 \times 10^{-7}}$	1.45081579
Average solution	**48.119030**	48.119838

define $X_i = [x_{i1} \quad x_{i2} \cdots x_{iD}]$
Initialize PSO-CREV
$n = 1$
do
  for i = 1 to the swarm size
    Calculate the fitness
    if $F(X_i(n)) < F(P_i^r(n))$
      $P_i^r(n) = X_i(n)$
    end if
  end for
  for i = 1 to the swarm size
    $P_i^g(n) = \arg\min_{j \in \Omega_i}(P_j^r(n))$
    Determine $w_i(n)$ using (25)
    $\xi_i(n) = w_i(n)\overline{\xi}_i(n)$
    Update $v_i(n+1)$ using (8)
    Clamp $v_i(n+1)$ within the range of $v_{\max}$
    Update $X_i(n+1)$ using (8)
  end for
  $n = n+1$
while stopping condition is fail or n < the maximal iteration

Fig. 4. Pseudocode for PSO-CREV algorithm.

adopted in Benchmark 1. All experiments are run for 25 times. The results reported are the averages (of the average fitness value in the swarm) calculated from all 25 runs. For each benchmark, the sizes of swarm are chosen as 10, 20, and 30, respectively.

A. Test Functions

Based on the popularity in the PSO community, the following functions are selected to allow for easier comparison. The details of these functions can be found in [24]. To compare the performance of all PSO algorithms under different circumstances, we will propose three benchmarks to test general convergence and robustness, and solve the hybrid composition function, respectively. All functions used are listed below, where o represents the shifted global optima and R represents an orthogonal matrix to rotate a particle's coordinate, whose values can be referred to [24]. The interval of X represents the range of distribution of particles at the beginning. During the optimization process, there is no bound for particle coordinates.

Benchmark 1: Test for Behavior of Convergence

- Shifted Sphere function, $X \in [-100 \;\; 100]^D$

$$F_{11}(X) = \sum_{i=1}^{D} z_i^2 \qquad Z = X - o.$$

- Shifted Griewank's function, $X \in [0 \;\; 600]^D$

$$F_{12}(X) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \qquad Z = X - o.$$

- Shifted Ackley's function, $X \in [-32 \;\; 32]^D$

$$F_{13}(X) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + 20 + e \qquad Z = X - o.$$

- Shifted Rastrigin's function, $X \in [-5 \;\; 5]^D$

$$F_{14} = \sum_{i=1}^{D} \left(z_i^2 - 10\cos(2\pi z_i) + 10\right) \qquad Z = X - o.$$

- Shifted Rotated Rastrigin's function $X \in [-5 \;\; 5]^D$

$$F_{15} = \sum_{i=1}^{D} \left(Z_i^2 - 10\cos(2\pi z_i) + 10\right) \qquad Z = (X - o) \cdot R.$$

Benchmark 2: Test for Robustness Under Disturbances

- Shifted Schwefel's problem, $X \in [-100 \;\; 100]^D$

$$F_{21} = \sum_{i=1}^{D} \left(\sum_{j=i}^{i} z_j\right)^2 \qquad Z = X - o.$$

- Shifted Schwefel's problem with noise in fitness, $X \in [-100 \;\; 100]^D$

$$F_{22} = \left(\sum_{i=1}^{D}\left(\sum_{j=i}^{i} z_j\right)^2\right) \cdot (1 + 0.4|N(0,1)|) \qquad Z = X - o.$$

Benchmark 3: Test for Continuous and Noncontinuous Hybrid Composition Functions

- Continuous version of hybrid function, $X \in [-5 \;\; 5]^D$.

  Employing the definitions of expanded function and composition function defined in [24], we define a hybrid composition function as

$$F_{31}(X) = \sum_{i=1}^{5} \left\{\eta_i \cdot C \cdot \frac{f_i(X/\lambda_i)}{f_i^{\max}}\right\}$$

where $C = 2000$ and η_i represents a kind of weight value in terms of

$$\eta_i = \exp\left(-\frac{\sum_{k=1}^{D} x_k^2}{2D\sigma_i^2}\right)$$

where $\sigma = [1 \quad 1 \quad 1 \quad 2 \quad 2]$, $\lambda = [0.25 \quad 5 \quad 5 \quad 50 \quad 0.125]$, $f_i^{\max} = f_i(Y/\lambda_i)$, where $Y = [5 \quad 5 \quad \cdots \quad 5]^D$, and $f_i(X)$ represents the basic functions used to construct the composition function, which includes the following five functions.

Expanded Scaffer's Function

$$\begin{aligned} g(x,y) &= 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2} \\ f_1(X) &= g(x_1,x_2) + g(x_2,x_3) \\ &\quad + \cdots + g(x_{D-1},x_D) + g(x_d,x_1). \end{aligned}$$

Rastrigin's Function

$$f_2 = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right).$$

Composition Function

$$\begin{aligned} g_1(X) &= \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ g_2(X) &= \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1}^2\right)^2 + (x_i - 1)^2\right) \\ f_3(X) &= g_1\left(g_2(x_1,x_2)\right) + g_1\left(g_2(x_3,x_4)\right) + \cdots \\ &\quad + g_1\left(g_2(x_{D-1},x_D)\right) + g_1\left(g_2(x_D,x_1)\right). \end{aligned}$$

Weierstrass Function

$$\begin{aligned} f_4 = \sum_{i=1}^{D} \left(\sum_{k=0}^{k_{\max}} \left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)\right]\right) \\ - D\sum_{k=0}^{k_{\max}} \left[a^k \cos(2\pi b^k \cdot 0.5)\right]. \end{aligned}$$

Griewank's Function

$$f_5(X) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

- Noncontinuous version of the hybrid function

The noncontinuous version of the hybrid function denoted by $F_{32}(X)$ is almost the same as the continuous one, except that

$$x_j = \begin{cases} x_j, & |x_j| < 1/2 \\ \text{round}(2x_j)/2, & |x_j| \geq 1/2 \end{cases}$$

for $j = 1, 2, \ldots, D$, where

$$\text{round}(x) = \begin{cases} p-1, & x \leq 0 \text{ and } q \geq 0.5 \\ p, & q < 0.5 \\ p+1, & x > 0 \text{ and } q \geq 0.5 \end{cases}$$

where P is x's integral part and q is x's decimal part.

TABLE II
RANGE OF $\bar{\xi}$ USED IN BENCHMARKS

Function	Range of $\bar{\xi}$
F_{11}	$[-0.5 \quad 0.5]^D$
F_{12}	$[-10 \quad 10]^D$
F_{13}	$[-2.5 \quad 2.5]^D$
F_{14}, F_{15}	$[-5 \quad 5]^D$
F_{21}, F_{22}	$[-5 \quad 5]^D$
F_{31}, F_{32}	$[-0.25 \quad 0.25]^D$

TABLE III
COMPARATIVE RESULTS OF F_{11} FOR BENCHMARK 1

Shifted Sphere Function (F_{11})				
D	Algorithm	M	Average	Std. Dev.
30	PSO-CREV ($\xi \neq 0$)	10	0.3822×10^{-7}	0.8270×10^{-9}
		20	0.3568×10^{-7}	0.4622×10^{-9}
		30	0.3562×10^{-7}	0.2948×10^{-9}
	PSO-CREV ($\xi = 0$)	10	129.42	31.4499
		20	0.1184	0.08889
		30	0.2284×10^{-4}	0.2010×10^{-4}
	PSO-LDIW	10	0.7366×10^{-2}	0.4628×10^{-2}
		20	0.1764×10^{-3}	0.5127×10^{-4}
		30	0.1414×10^{-3}	0.2252×10^{-4}
	GCPSO	10	0.3544×10^{-7}	0.1899×10^{-7}
		20	0.1049×10^{-19}	0.3882×10^{-20}
		30	0.1800×10^{-22}	0.3739×10^{-23}
	MPSO-TVAC	10	0.4410	0.1248
		20	0.2112	0.07165
		30	0.1834	0.05884
	CPSO-H_6	10	**0**	**0**
		20	**0**	**0**
		30	**0**	**0**
	PSO-DVM	10	0.02688	0.1670×10^{-2}
		20	0.01354	0.6385×10^{-3}
		30	0.01126	0.4183×10^{-3}
100	PSO-CREV ($\xi \neq 0$)	10	0.2388×10^{-3}	0.6690×10^{-5}
		20	0.1272×10^{-3}	0.1610×10^{-5}
		30	0.1105×10^{-3}	0.1523×10^{-5}
	PSO-CREV ($\xi = 0$)	10	0.3443×10^{5}	0.2030×10^{4}
		20	0.4439×10^{4}	0.3631×10^{3}
		30	0.6652×10^{3}	70.6634
	PSO-LDIW	10	0.3077×10^{4}	0.2168×10^{3}
		20	0.1930×10^{4}	0.1464×10^{3}
		30	0.2102×10^{4}	0.1016×10^{3}
	GCPSO	10	0.2803×10^{4}	0.2729×10^{3}
		20	10.0099	4.5058
		30	0.2178	0.1241
	MPSO-TVAC	10	0.7718×10^{3}	0.1672×10^{3}
		20	8.7721	5.4689
		30	0.1417	0.02909
	CPSO-H_6	10	$\mathbf{0.4333 \times 10^{-13}}$	$\mathbf{0.4051 \times 10^{-13}}$
		20	$\mathbf{0.6957 \times 10^{-29}}$	$\mathbf{0.4396 \times 10^{-29}}$
		30	$\mathbf{0.3433 \times 10^{-30}}$	$\mathbf{0.1476 \times 10^{-30}}$
	PSO-DVM	10	2.4943	0.08823
		20	0.7671	0.02511
		30	0.4322	0.01150

TABLE IV
COMPARATIVE RESULTS BETWEEN F_{12} AND F_{13} FOR BENCHMARK 1

			Shifted Griewank's Function F_{12}		Shifted Ackley's Function F_{13}	
Dimension	Algorithm	M	Average	Std. Dev.	Average	Std. Dev.
30	PSO-CREV ($\xi \neq 0$)	10	$\mathbf{0.4773 \times 10^{-2}}$	$\mathbf{0.1339 \times 10^{-2}}$	0.7870×10^{-2}	0.8474×10^{-4}
		20	$\mathbf{0.3087 \times 10^{-2}}$	$\mathbf{0.8008 \times 10^{-3}}$	0.7706×10^{-2}	0.5292×10^{-4}
		30	$\mathbf{0.4238 \times 10^{-3}}$	$\mathbf{0.2898 \times 10^{-3}}$	0.7702×10^{-2}	0.3204×10^{-4}
	PSO-CREV ($\xi = 0$)	10	1.3279	0.07743	6.3440	0.3411
		20	0.03288	0.5504×10^{-2}	2.2372	0.1467
		30	0.6799×10^{-2}	0.2013×10^{-2}	0.9780	0.1211
	PSO-LDIW	10	0.02522	0.4817×10^{-2}	1.1166	0.2218
		20	0.03171	0.01331	0.1547	0.09573
		30	0.1018	0.02810	0.05312	0.02963
	GCPSO	10	0.02296	0.5647×10^{-2}	4.7501	0.4442
		20	0.01053	0.1832×10^{-2}	1.4912	0.3776
		30	0.8854×10^{-2}	0.2581×10^{-2}	0.1673	0.09076
	MPSO-TVAC	10	0.09421	0.01515	0.04139	0.9151×10^{-2}
		20	0.04482	0.01186	0.01557	0.4019×10^{-2}
		30	0.02331	0.5701×10^{-2}	0.01098	0.2753×10^{-2}
	CPSO-H_6	10	0.02780	0.4905×10^{-2}	$\mathbf{0.2100 \times 10^{-13}}$	$\mathbf{0.1217 \times 10^{-14}}$
		20	0.01545	0.2843×10^{-2}	$\mathbf{0.1517 \times 10^{-13}}$	$\mathbf{0.6702 \times 10^{-15}}$
		30	0.5051×10^{-2}	0.1601×10^{-2}	$\mathbf{0.1418 \times 10^{-13}}$	$\mathbf{0.6122 \times 10^{-15}}$
	PSO-DVM	10	0.05893	0.01528	0.04048	0.1107×10^{-2}
		20	0.03156	0.01594	0.02655	0.8484×10^{-3}
		30	0.04467	0.01373	0.02313	0.4086×10^{-3}
100	PSO-CREV ($\xi \neq 0$)	10	$\mathbf{0.4985 \times 10^{-2}}$	$\mathbf{0.5334 \times 10^{-3}}$	**0.5612**	0.09864
		20	$\mathbf{0.9000 \times 10^{-3}}$	$\mathbf{0.3463 \times 10^{-4}}$	**0.01092**	$\mathbf{0.8576 \times 10^{-3}}$
		30	$\mathbf{0.5526 \times 10^{-3}}$	$\mathbf{0.1840 \times 10^{-4}}$	0.8239×10^{-2}	0.8251×10^{-3}
	PSO-CREV ($\xi = 0$)	10	94.2509	5.1210	18.2851	0.1874
		20	12.7656	0.9140	11.9619	0.2014
		30	2.4755	0.1783	7.7433	0.2737
	PSO-LDIW	10	8.2507	0.6568	10.9544	0.2698
		20	5.5343	0.2759	8.5619	0.1802
		30	5.2790	0.2697	6.7394	0.6873
	GCPSO	10	27.7644	5.4501	16.0869	0.2845
		20	0.6541	0.06927	6.7431	0.4506
		30	0.06372	0.02649	2.6214	0.2917
	MPSO-TVAC	10	3.5804	0.4502	7.8697	0.2582
		20	0.2967	0.04796	3.6326	0.1778
		30	0.04282	0.01186	1.7673	0.2030
	CPSO-H_6	10	0.02433	0.5916×10^{-2}	2.8965	**0.08422**
		20	0.3844×10^{-2}	0.1098×10^{-2}	0.4566	0.1245
		30	0.3938×10^{-2}	0.1493×10^{-2}	$\mathbf{0.4022 \times 10^{-15}}$	$\mathbf{0.4210 \times 10^{-16}}$
	PSO-DVM	10	0.4578	0.01243	0.8141	0.05648
		20	0.2404	0.7072×10^{-2}	0.1549	0.4491×10^{-2}
		30	0.1476	0.4464×10^{-2}	0.1050	0.3947×10^{-2}

B. PSO Configurations

The configurations of all algorithms chosen for comparison are introduced as follows.

- PSO-CREV ($\xi \neq 0$): The updating principle is of the form (8) with strategy of $\varepsilon(n)$ in the form of (26) and a decreasing $\xi(n)$ whose $w(n)$ is in the form of (25). All parameters are chosen as follows: $c_1 = c_2 = 3.5$, $\alpha = 0.95$, $a = 3.5$, $b = 0.35$, $\eta = 0.993$. The ranges of $\bar{\xi}(n)$ are chosen with respect to different fitness functions, just as Table II shows. Here, the ranges of $\bar{\xi}$ are chosen by trail and error ways.
- PSO-CREV ($\xi = 0$): All parameters are the same as PSO-CREV ($\xi \neq 0$) except that $\xi = 0$.
- PSO-LDIW: The typical PSO with linearly decreasing inertial weight (LDIW) [25] is selected, where the inertial weight is varied from 0.9 at the beginning of search to 0.4 at the end, and $c_1 = c_2 = 2$.
- GCPSO: The details of GCPSO can be found in [22]. In addition, the parameters are chosen as $c_1 = c_2 = 1.49$; the inertial weight is 0.72. A dynamic strategy of f_c and s_c is adopted where $f_c = 5$ and $s_c = 15$ initially.
- MPSO-TVAC: The details of MPSO-TVAC can be found in [16]. According to the presentation in [16], the best ranges for c_1 and c_2 are $2.5 \rightarrow 0.5$ and $0.5 \rightarrow 2.5$, respectively. The inertial weight is set to change from 0.9 to 0.4 over the iterations. The mutation step size is set to change from $V_{\max}$ to $0.1V_{\max}$ over the search.
- CPSO-H_6: The details of CPSO can be found in [23]. Here, we split solution vector into six parts. The updating principle with decreasing inertial weight is used, where the inertial weight decreases from 0.9 to 0.4 over the search, $c_1 = c_2 = 1.49$.
- PSO-DVM: The typical PSO-DVM [21] is selected, where the inertial weight is fixed as 1, and $c_1 = c_2 = 2$. The clamped velocity $V_{\max}$ is decreasing from the upper limit of the initial bound of the particle coordinates according to the principle $V_{\max}(n+1) = (1 - (n/N_b)^{0.05})V_{\max}(n)$, where N_b represents the total iterations and n is the current iteration.

In addition, the maximum velocity $V_{\max}$ for all algorithms except PSO-DVM is set to the upper limit of the initial bound of the particles' coordinates. Also, for all algorithms, the best square version of neighborhood is used [26], [27].

VI. RESULTS

All results of benchmarks are displayed in Tables III–VII, in which the "Average" column represents the average fitness values calculated from 25 runs. The column labeled by "Std. Dev." shows the standard deviation of fitness values resulting

TABLE V
COMPARATIVE RESULTS BETWEEN F_{14} AND F_{15} FOR BENCHMARK 1

			Shifted Rastrigin's Function F_{14}		Shifted Rotated Rastrigin's Function F_{15}	
Dimension	Algorithm	M	Average	Std. Dev.	Average	Std. Dev.
30	PSO-CREV ($\xi \neq 0$)	10	64.2387	4.3620	0.1391×10^3	11.2118
		20	59.2234	2.6873	0.1553×10^3	6.2396
		30	60.9137	3.5207	0.1522×10^3	4.1701
	PSO-CREV ($\xi = 0$)	10	91.4731	5.9383	0.2105×10^3	13.0574
		20	97.8075	9.2203	0.1759×10^3	11.1724
		30	0.1168×10^2	6.5667	0.1890×10^3	6.5545
	PSO-LDIW	10	0.1043×10^3	9.4587	0.3594×10^3	6.5197
		20	0.1412×10^3	7.7136	0.3649×10^3	4.9723
		30	0.1663×10^3	4.7726	0.3671×10^3	4.7231
	GCPSO	10	0.1192×10^3	5.5806	0.2395×10^3	13.3916
		20	0.1038×10^3	8.3613	0.2627×10^3	10.7929
		30	0.1279×10^3	10.0482	0.2638×10^3	9.2072
	MPSO-TVAC	10	40.5186	3.1625	0.2967×10^3	14.5016
		20	76.2267	7.3724	0.2063×10^3	12.3903
		30	93.4475	5.2427	0.1635×10^3	8.7490
	CPSO-H_6	10	**5.2722**	**0.4639**	$\mathbf{0.1076 \times 10^3}$	**6.1484**
		20	**1.8479**	**0.3876**	**91.3767**	**3.3922**
		30	**1.4526**	**0.3872**	**83.2181**	**3.1833**
	PSO-DVM	10	72.3064	7.4676	0.1763×10^3	10.5593
		20	0.1304×10^2	7.8374	0.2073×10^3	6.4866
		30	0.1540×10^2	5.7099	0.2053×10^3	3.2621
100	PSO-CREV ($\xi \neq 0$)	10	0.3181×10^3	9.5209	$\mathbf{0.8971 \times 10^3}$	38.4168
		20	0.2631×10^3	16.1141	$\mathbf{0.7599 \times 10^3}$	25.7992
		30	0.3542×10^3	21.8051	$\mathbf{0.7875 \times 10^3}$	13.6576
	PSO-CREV ($\xi = 0$)	10	0.7176×10^3	19.7917	0.2074×10^4	46.0470
		20	0.5242×10^3	24.4770	0.1450×10^4	35.2493
		30	0.5110×10^3	26.1451	0.1402×10^4	30.5256
	PSO-LDIW	10	0.5511×10^3	21.4084	0.2190×10^4	54.2470
		20	0.7095×10^3	34.3618	0.2206×10^4	45.9512
		30	0.7637×10^3	22.0664	0.2528×10^4	30.9686
	GCPSO	10	0.3904×10^3	13.7100	0.1815×10^4	61.1590
		20	0.4329×10^3	27.1632	0.1442×10^4	39.1926
		30	0.5113×10^3	21.0679	0.1423×10^4	30.8984
	MPSO-TVAC	10	0.3081×10^3	10.6992	0.1276×10^4	38.6419
		20	0.3256×10^3	16.1550	0.1158×10^4	34.1001
		30	0.3974×10^3	24.9777	0.1070×10^4	38.3926
	CPSO-H_6	10	$\mathbf{0.1890 \times 10^3}$	**5.0082**	0.1409×10^4	50.8619
		20	$\mathbf{0.1282 \times 10^3}$	**3.2454**	0.1044×10^4	42.4373
		30	$\mathbf{0.1085 \times 10^3}$	**2.9348**	0.8931×10^3	39.1990
	PSO-DVM	10	0.2799×10^3	20.6961	0.1006×10^4	**36.6810**
		20	0.3771×10^3	31.5027	0.8975×10^3	**22.1005**
		30	0.4772×10^3	16.5687	0.8878×10^3	**12.9851**

TABLE VI
COMPARATIVE RESULTS BETWEEN F_{21} AND F_{22} FOR BENCHMARK 2

			Shifted Schwefel's Problem F_{21}		Shifted Schwefel's Problem with Noise F_{22}	
Dimension	Algorithm	M	Average	Std. Dev.	Average	Std. Dev.
30	PSO-CREV ($\xi \neq 0$)	10	14.8728	1.7839	0.4847×10^4	0.1181×10^4
		20	2.0672	0.3163	0.1865×10^3	62.6635
		30	0.8326	0.1053	24.1814	4.6605
	PSO-CREV ($\xi = 0$)	10	0.1802×10^5	0.9476×10^3	0.5176×10^5	0.3604×10^4
		20	0.5848×10^4	0.6090×10^3	0.2823×10^5	0.1708×10^4
		30	0.2780×10^4	0.2783×10^3	0.1796×10^5	0.1246×10^4
	PSO-LDIW	10	0.2021×10^4	0.1968×10^3	0.7828×10^4	0.9574×10^3
		20	61.3987	8.2595	0.1942×10^4	0.2718×10^3
		30	24.4172	8.4681	0.9200×10^3	0.1241×10^3
	GCPSO	10	0.1704×10^4	0.2330×10^3	0.1468×10^5	0.1772×10^4
		20	31.9135	16.6248	0.4392×10^3	91.1818
		30	0.5837	0.1589	0.1370×10^3	33.8733
	MPSO-TVAC	10	0.3513×10^3	40.7070	0.4112×10^5	0.5436×10^4
		20	36.2835	3.9492	0.2644×10^5	0.4855×10^4
		30	27.6208	6.4923	0.1615×10^5	0.2157×10^4
	CPSO-H_6	10	$\mathbf{0.7557 \times 10^{-7}}$	$\mathbf{0.3240 \times 10^{-7}}$	**0.01622**	$\mathbf{0.4238 \times 10^{-2}}$
		20	$\mathbf{0.1544 \times 10^{-13}}$	$\mathbf{0.8474 \times 10^{-14}}$	$\mathbf{0.8189 \times 10^{-5}}$	$\mathbf{0.2380 \times 10^{-5}}$
		30	$\mathbf{0.7493 \times 10^{-16}}$	$\mathbf{0.2132 \times 10^{-16}}$	$\mathbf{0.2746 \times 10^{-6}}$	$\mathbf{0.7044 \times 10^{-7}}$
	PSO-DVM	10	14.2836	0.9712	0.9327×10^4	0.1230×10^4
		20	4.9295	0.3866	0.1629×10^4	0.2779×10^3
		30	3.1522	0.1991	0.4445×10^3	71.3235

from 25 runs. The best result in a comparison is emphasized as bold font.

It should be pointed out that in each run, the fitness value at each iteration is the average of fitness values with respect to all particles' current positions, i.e., Fitness$(n) = 1/M \sum_{i=1}^{M} F(X_i(n))$. That is different from the fitness referred in many literatures, where at each iteration, it results from the global best solution found so far, i.e.,

TABLE VII
COMPARATIVE RESULTS BETWEEN F_{31} AND F_{32} FOR BENCHMARK 3

			Continuous Hybrid Composition Functions F_{31}		Non-continuous Hybrid Composition Functions F_{32}	
Dimension	Algorithm	M	Average	Std. Dev.	Average	Std. Dev.
30	PSO-CREV ($\xi \neq 0$)	10	2.4526	1.0499	1.5114	0.3187
		20	0.1509	0.2062×10^{-2}	0.2769	0.02671
		30	0.1470	0.9082×10^{-3}	0.2466	0.1265×10^{-2}
	PSO-CREV ($\xi = 0$)	10	33.3237	2.2770	27.3119	1.2117
		20	7.7710	0.7079	6.8764	0.6470
		30	2.3756	0.3312	2.3868	0.4066
	PSO-LDIW	10	2.0952	0.2562	3.2042	0.5074
		20	0.5075	0.1595	0.5366	0.1427
		30	0.1360	0.01662	0.1625	0.01270
	GCPSO	10	31.8032	2.6650	23.4257	1.9970
		20	3.6694	0.7161	2.7865	0.3530
		30	0.7960	0.2420	0.4824	0.1265
	MPSO-TVAC	10	0.2491	0.03339	0.2947	0.06744
		20	0.08492	0.01589	0.08187	0.01584
		30	0.4125	0.3637	0.05059	0.7077×10^{-2}
	CPSO-H_6	10	$\mathbf{0.3699 \times 10^{-3}}$	$\mathbf{0.1192 \times 10^{-3}}$	$\mathbf{0.8482 \times 10^{-2}}$	$\mathbf{0.7612 \times 10^{-2}}$
		20	$\mathbf{0.1806 \times 10^{-3}}$	$\mathbf{0.2781 \times 10^{-5}}$	$\mathbf{0.1808 \times 10^{-3}}$	$\mathbf{0,3095 \times 10^{-5}}$
		30	$\mathbf{0.1780 \times 10^{-3}}$	$\mathbf{0.4386 \times 10^{-5}}$	$\mathbf{0.1815 \times 10^{-3}}$	$\mathbf{0.3706 \times 10^{-5}}$
	PSO-DVM	10	3.5366	0.1171	3.3572	0.1067
		20	2.3804	0.04215	2.3783	0.04273
		30	2.1291	0.03304	2.0866	0.02772

Fitness$(n) = \min_{1 \le i \le M} F(P_i^r(n))$. Since the more important concern of this paper is not only to test the PSO performance but also to investigate the aggregation among particles, obviously, Fitness$(n) = 1/M \sum_{i=1}^{M} F(X_i(n))$ reflects not only the best fitness found so far but also the distribution of particles—if PSO does not converge enough or some particles are trapped into local minima, the fitness will be pulled larger. But, if particles converge enough, it holds that Fitness$(n) = 1/M \sum_{i=1}^{M} F(X_i(n)) \approx F(X_i(n)) \approx \min_{1 \le i \le M} F(P_i^r(n))$. Hence, this average fitness is proper to reflect the actual performance of PSO.

A. Results of Benchmark 1

The aim of Benchmark 1 is to test the performances of all PSOs in solving single optimization function without disturbance. The results are shown in Tables III–V.

1) Shifted Sphere Function: The shifted sphere function is a simple unimodal function which has a unique global optimum without any local minima. Hence, this function is proposed to test convergent rates of all algorithms. Table III shows the results of all tests using seven PSO algorithms. In addition, Fig. 5 shows the average function evolutions over iterations, where "$(N = 30)$" denotes that the swarm size is 30. The plots of evolutional processes with respect to other two population sizes are similar, except that the final results over search are different.

Obviously, when dimension is taken as 30, CPSO-H_6 converges more quickly than other algorithms. In particular, it is very impressive that, with respect to swarm size of 30, CPSO-H_6 uses a little more than half a run to converge into the vicinity below 10^{-30}. GCPSO takes the second place, which performed better and better with increasing swarm size. Following it, PSO-CREV performs better than other algorithms. But, when the dimension of the solution space is chosen as 100, PSO-CREV ($\xi \neq 0$) shows its high exploration ability, so that it is one of the two algorithms with optimized fitness values under 0.01.

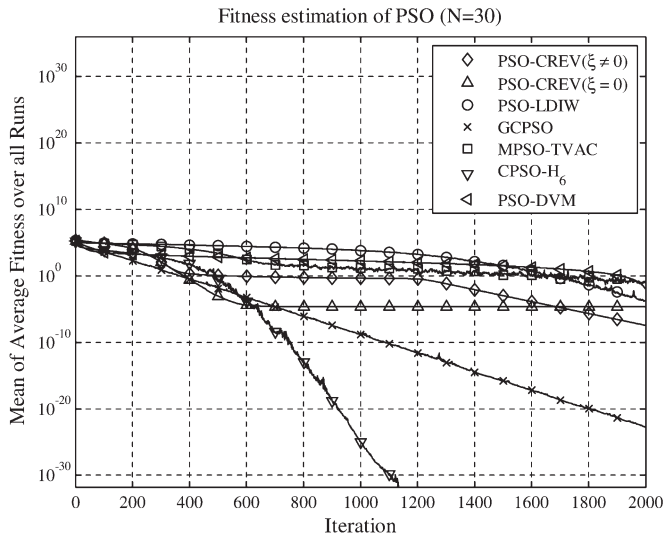

Fig. 5. Average variation of the optimum value of shifted sphere function over iterations, where the dimension of solution space is 30.

A significant character of PSO-CREV is observed that there is a relatively little difference about the performance of PSO-CREV ($\xi \neq 0$) with respect to different swarm sizes, while the performance of other algorithms, except CPSO-H_6, is more sensitive to the swarm size. This phenomenon implies that the key factor in determining the PSO-CREV performance is not the swarm size, but the random velocity $\xi(n)$. Hence, given an optimization task, PSO-CREV using small population size performs as good as, even better than other PSO algorithms, which will be verified in the following test.

Fig. 5 shows the estimation process of fitness over iterations. Another character of PSO-CREV is observed that, from nearly 500 iterations to 1200 iterations [exactly the 60% of total iterations which is the threshold in (25)], the fitness values over iterations are kept almost the same. It looks like a kind of stagnation. To explain this phenomenon, we first propose the convergence process of all particles of PSO-CREV in Fig. 6, which shows the distribution of particles in the first dimension

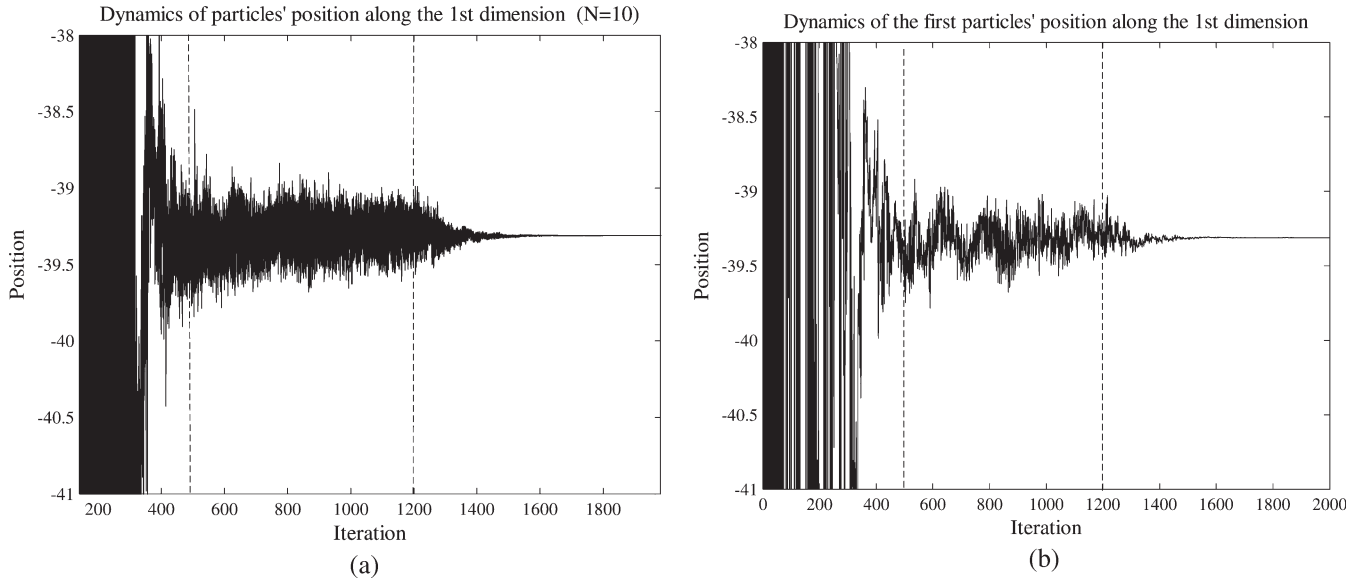

Fig. 6. (a) All particles positions over iterations along the first dimension. To clearly illustrate the dynamics of a single particle, the (b) positions of the first particle are shown as an example. To emphasize the impact brought by $\xi(n)$, the bound of plot is set to $[-41, -38]$.

of the solution space. Obviously, within a certain short initial period, particles are going to aggregate together. But, when they approach the best solution after nearly 500 iterations, the cognitive component and social component have become so small relative to the bound of random velocity $\xi(n)$, that $\xi(n)$ dominates the behavior of particles. Hence, particles behave like random exploration around the best solution, i.e., they distributed uniformly around the best solution. Due to the average fitness formula mentioned above, the average fitness of the swarm is kept almost the same during this period. That is why PSO-CREV seems to be stagnated. But, in fact, they randomly search the vicinity of the best solution. After 1200 iterations, due to (25), the bound of $\xi(n)$ becomes smaller and smaller, so that particles converge to the best solution quickly, just as the plot of fitness of PSO-CREV is indicated in Fig. 5, where the fitness is shown as a straight line after 1200 iterations. It should be noted that, although there is no distinct change about PSO-CREV's average fitness from 500 to 1200 iterations, the average fitness is indeed decreased over iterations very slowly because of decreasing $\varepsilon(n)$.

2) Shifted Griewank's Function and Shifted Ackley's Function: Comparing with the sphere function, it is more difficult to optimize Griewank's function and Ackley's function, because both functions are multimodal. That means there are multiple local optimal solutions around the global best solution. From the results of both tests shown in Table IV, the performance of PSO-CREV with nonzero ξ is very impressive in solving this kind of multimodal function.

The main character of PSO-CREV can be observed from the results. Comparing with PSO-CREV $(\xi = 0)$, PSO-CREV $(\xi \neq 0)$ behaves itself with strong exploration ability, which is brought by nonzero ξ, so that with the same population size, PSO-CREV $(\xi \neq 0)$ is more active to explore the solution space than other algorithms. Hence, the results of PSO-CREV $(\xi \neq 0)$ are much better than most of the other algorithms, even better than CPSO-H_6 in the tests on Griewank's function, where

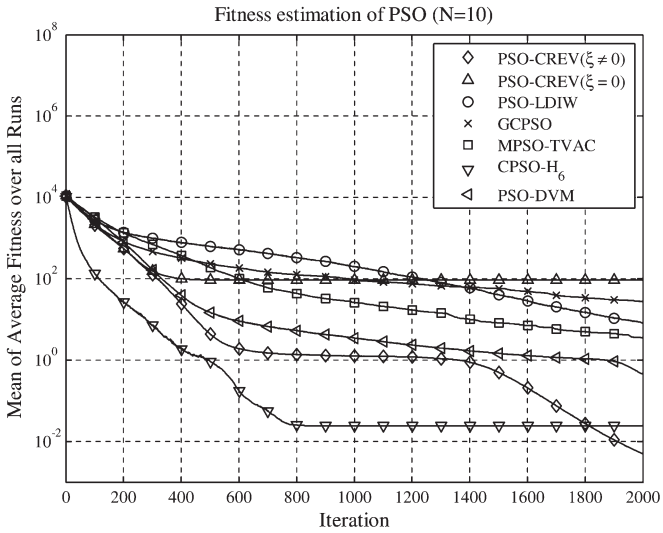

Fig. 7. Average variation of fitness value in the test of Griewank's function $(F_{12}(X))$, where the population size is ten, and the dimension of solution space is 100.

the dimension of the solution space is set to 100, just as Fig. 7 shows, where the function evaluations in terms of population size 10 are depicted. Obviously, even if the population size is far less than the solution space's dimension, PSO-CREV $(\xi \neq 0)$ will not stagnate at the end of the process. At the same time, for three different population sizes, the results of PSO-CREV $(\xi \neq 0)$ in the tests on Ackley's function with dimension of 30 are almost the same. That implies that in this test, it is $\xi(n)$, but not the swarm size, who plays the dominant role to determine the convergent speed of PSO-CREV.

3) Shifted Rastrigin's Function and Shifted Rotated Rastrigin's Function: The main character of the tests on Rastrigin's function is that the local optima's number is so huge that particles are easily stagnated in local optima.

Table V shows the results of Rastrigin's function. A common phenomenon observed is that as population size is increased, the best fitness values optimized by some algorithms are also increased. This is apparently an unreasonable phenomenon resulted from the formula of fitness, $\text{Fitness}(n) = 1/M \sum_{i=1}^{M} F(X_i(n))$. Due to a large number of closed local minima, some particles are trapped in different local minima. At the same time, because of the interaction among particles, $P^r(n)$ and $P^g(n)$ recorded by them are not the same, so that these particles will swing between their $P^r(n)$ and $P^g(n)$ due to $r_1(n)$ and $r_2(n)$. Thus, the fitness values related to these particles will be relatively higher. Therefore, the higher a population size is, the more particles swing, and the higher the average fitness $\text{Fitness}(n)$ is.

For shifted-only version of Rastrigin's function, MPSO-TVAC, PSO-CREV ($\xi \neq 0$), and CPSO-H_6 can optimize fitness value below 100 over 2000 iterations. Undoubtedly, due to cooperative swarms, CPSO-H_6 takes the first place in both tests. Also, the second winner is PSO-CREV, followed by MPSO-TVAC.

For the shifted rotated version, the important property is verified again, that the performance of PSO-CREV is mostly determined by the random velocity $\xi(n)$ but not the population size. Moreover, with increasing population size, the performance of PSO-CREV becomes more and more steady, and the standard deviation is decreased.

B. Results of Benchmark 2

Benchmark 2 is proposed to compare the performance of all PSO algorithms under the circumstance that there is disturbance in fitness function. Table VI shows the results of the tests on shifted Schwefel's problem with and without disturbance. From the results about the nondisturbance Schwefel's problem, it is observed that CPSO-H_6 is more powerful to find the global optimum than other PSOs.

As a comparison, when the stochastic noise is added to Schwefel's problem, the fitness values optimized by PSO-CREV ($\xi \neq 0$) can be kept below 200, while only CPSO-H_6 performs better than it.

C. Results of Benchmark 3

The aim of Benchmark 3 is to test the PSOs' performance when the fitness function is a hybrid composition function. Moreover, a noncontinuous version is also employed to compare the PSOs' ability in solving noncontinuous optimization problems.

The results of Benchmark 3 are shown in Table VII. The first phenomenon observed from the table is that, for every algorithm, there is not too much difference between the performance in the tests of the continuous and noncontinuous versions of the hybrid function. Therefore, the property of noncontinuity does not affect the performance of PSOs too much. Second, it is impressive that CPSO-H_6 performs much better than other algorithms, which optimizes the fitness value less than 0.01 over 2000 iterations. Besides CPSO-H_6, the performance of MPSO-TVAC is good, which almost beats PSO-CREV for all population sizes, except population size of 30 in a continuous version of the hybrid function, where the final fitness is 0.4125, larger than 0.1470 that resulted from PSO-CREV ($\xi \neq 0$). For MPSO-TVAC, when particles are closed to the best solution, there are many opportunities for MPSO-TVAC with large population size to execute a mutation operation. Hence, the average fitness of MPSO-TVAC may be often suddenly increased. Hence, there is relatively high probability that the sudden increasing occurs at the end of the search (2000 iterations). That is why, with high population size, MPSO-TVAC performs worse than that with small population size. For PSO-CREV ($\xi \neq 0$), it can be observed that when population sizes are 20 and 30, the difference of PSO-CREV ($\xi \neq 0$) performance is very small relative to other PSO algorithms. That implies that for PSO-CREV ($\xi \neq 0$), the most important factor affecting its performance is the random velocity $\xi(n)$, but not the population size.

TABLE VIII
COMPARATIVE RESULTS ON COMPUTATIONAL TIME

	Average Computational Time
PSO-CREV($\xi \neq 0$)	1
PSO-LDIW	0.9829
GCPSO	1.0095
MPSO-TVAC	0.9978
CPSO-H_6	6.3964
PSO-DVM	0.9841

Since benchmark 3 involves a lot of computations to evaluate fitness at one iteration, it is necessary to compare the computational cost of all algorithms. Table VIII shows the average time for one run of the optimization process, which results from 25 runs of the test on the continuous version, where the population size is set to 30. In Table VIII, the computational time of PSO-CREV ($\xi \neq 0$) is set to 1 unit, so that the computational times of the other algorithms are scaled relative to PSO-CREV ($\xi \neq 0$). Obviously, except CPSO-H_6, there are no more differences between computational times of other algorithms. Hence, relative to CPSO-H_6, which increases the computational time greatly, PSO-CREV looks like a more "economical" way to enhance the PSO performance.

D. Summary on Benchmarks

From the benchmarks proposed above, the following characters of PSO-CREV can be observed.

1) The factor dominating the PSO-CREV performance is the additional velocity ξ, while the performance of the other algorithms is more sensitive to swarm size. Without this stochastic velocity, PSO-CREV has no advantage than the traditional PSO paradigm, just as the results of PSO-CREV ($\xi = 0$) indicate.
2) Because of the first character, the performance of PSO-CREV seems to be more steady with respect to different swarm sizes. With increasing population size, there is limited improvement on the PSO-CREV performance, while the performance of others is enhanced more distinctly.
3) Due to the exploration strategy presented in (25), the trajectory of fitness evaluation referred to PSO-CREV

$(\xi \neq 0)$ can be split into two parts. During the iterations involving decreasing $w(n)$, because the bound of $\xi(n)$ is reduced, and cognitive and social components dominate the convergence of PSO-CREV, the fitness value is decreased quickly.

Moreover, only PSO-CREV and PSO-LDIW are the algorithms based on the basic PSO conception, while the other four algorithms append some operations to the basic PSO updating principle in terms of mutation and cooperative swarms. That means, from the viewpoint of structure, PSO-CREV and PSO-LDIW make improvements at the same level. If only comparing PSO-CREV with PSO-LDIW, it is very clear that PSO-CREV $(\xi \neq 0)$ performs better than the typical PSO-LDIW. In fact, almost all additional operations developed so far, such as MPSO-TVAC, GCPSO, and CPSO, can be transplanted into PSO-CREV to improve its performance further. For example, cooperating with differential evolution operator, which is employed in DEPSO [19], the PSO-CREV's performance in Rastrigin's function (F_{14}) will be enhanced significantly, even better than CPSO-H_6.

VII. Conclusion

In this paper, a modified PSO, named as PSO-CREV, is proposed. Due to decreasing ε, PSO-CREV satisfies the regular stochastic recursion paradigm, so that its convergence is proved using the Lyapunov theory on stochastic process. To illustrate PSO-CREV with other significant improvements, three benchmarks are proposed. From the proof and benchmarks, some properties of PSO-CREV are found.

1) The particles in PSO-CREV may be diverged at the beginning. Once the iteration of evolution goes over a certain threshold N_k, the PSO-CREV starts to converge.
2) A random velocity $\xi(n)$ with zero expectation can be added to updating velocity to enhance the exploration ability while the convergence of PSO-CREV is guaranteed mathematically.
3) The most dominant factor for the performance of PSO-CREV is $\xi(n)$, but not swarm size.

There are still some open questions waiting for further study. Although it is proved that PSO-CREV converges with probability one, the proof does not ensure that PSO-CREV converges to the global best solution with probability one. At the same time, Assumption 5) of Theorem 1 is too strong, which may be replaced by a weakened one if some requirements of the distribution of local minima are specified. As mentioned in Remark 1, when $P^r(n) \neq P^g(n)$, particles will not converge to a position but a region. Hence, it will be better to use dynamic c_1 and c_2 to make particles behave an inclined exploration, just like the time-varying acceleration constant developed in MPSO-TVAC.

References

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. MHS*, Nagoya, Japan, 1995, pp. 39–43.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, 1995, pp. 1942–1948.

[3] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.

[4] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability," in *Proc. Int. Conf. Intell. Syst. Appl. Power Syst.*, Rio de Janeiro, Brazil, 1999, pp. 117–121.

[5] M. A. Abido, "Particle swarm optimization for multimachine power system stabilizer design," in *Proc. Power Eng. Soc. Summer Meeting*, 2001, pp. 1346–1351.

[6] L. Messerschmidt and A. P. Engelbrecht, "Learning to play games using a PSO-based competitive learning approach," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 280–288, Jun. 2004.

[7] Y. Li and X. Chen, "Mobile robot navigation using particle swarm optimization and adaptive NN," in *Proc. 1st Int. Conf. Nat. Comput.*, Changsha, China, *Lecture Notes in Computer Science*, vol. 3612. Berlin, Germany: Springer-Verlag, 2005, pp. 554–559.

[8] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[9] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. New York: Springer-Verlag, 2003.

[10] H. M. Emara and H. A. Fattah, "Continuous swarm optimization technique with stability analysis," in *Proc. Amer. Control Conf.*, 2004, vol. 3, pp. 2811–2817.

[11] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[12] K. Yasuda, A. Ide, and N. Iwasaki, "Adaptive particle swarm optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2003, pp. 1554–1559.

[13] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, 2003.

[14] B. Brandstäter and U. Baumgartner, "Particle swarm optimization—Mass-spring system analogon," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 997–1000, Mar. 2002.

[15] Y. Liu, Z. Qin, and Z. Shi, "Hybrid particle swarm optimizer with line search," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2004, vol. 4, pp. 3751–3755.

[16] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.

[17] R. Poli, C. D. Chio, and W. B. Langdon, "Exploring extended particle swarms: A genetic programming approach," in *Proc. Conf. Genet. and Evol. Comput.*, Washington, DC, 2005, pp. 169–176.

[18] K. E. Parsopouls and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Nat. Comput.*, vol. 1, no. 2/3, pp. 235–306, Jun. 2002.

[19] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Washington, DC, 2003, pp. 3816–3821.

[20] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, 1998, pp. 69–73.

[21] H. Y. Fan and Y. Shi, "Study on Vmax of particle swarm optimization," in *Proc. Workshop Particle Swarm Opt.*, Indianapolis, IN, 2001.

[22] F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, 2002.

[23] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[24] 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," IIT, Kanpur, India, NTU, Singapore and KanGAL Tech. Rep. 2 005 005, May 2005.

[25] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proc. 7th Conf. Evol. Program.*, New York, 1998, pp. 591–600.

[26] Y. Shi and R. Eberhart, "An empirical study of particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, 1999, pp. 1945–1949.

[27] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simple, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

Xin Chen received the B.Sc. degree in industrial automation and the M.Sc. degree in control theory and engineering from Central South University, Changsha, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree in electromechanical engineering at the University of Macau, Taipa, Macao S.A.R., China.

His research interests include swarm intelligence and multiple-robot coordination.

Yangmin Li (M'98–SM'04) received the B.Sc. and M.Sc. degrees in mechanical engineering from Jilin University, Changchun, China, in 1985 and 1988, respectively, and the Ph.D. degree in mechanical engineering from Tianjin University, Tianjin, China, in 1994.

After that, he was a Lecturer with the South China University of Technology, a Fellow with the International Institute for Software Technology of the United Nations University (UNU/IIST), a Visiting Scholar with the University of Cincinnati, and a Postdoctoral Fellow with Purdue University. He is currently an Associate Professor and Director of the Mechatronics Laboratory, Faculty of Science and Technology, University of Macau, Taipa, Macao S.A.R., China. He has authored or coauthored over 160 papers in international journals and conferences. He serves as an Editor of the *Chinese Journal of Mechanical Engineering*, a member of Editorial Consultant Board of the *International Journal of Advanced Robotic Systems*, and a Technical Editor of *Trends in Applied Sciences Research*.

Dr. Li is a member of the American Society of Mechanical Engineers.