

Review

A Lyapunov-Based Extension to Particle Swarm Dynamics for Continuous Function Optimization

Sayantani Bhattacharya ¹, Amit Konar ¹, Swagatam Das ¹ and Sang Yong Han ^{2,*}

¹ Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India; E-Mails: bhattacharya.sayantani@gmail.com (S.B.); konaramit@yahoo.co.in (A.K.); swagatamdas19@yahoo.co.in (S.D.)

² School of Computer Science and Engineering, Chung-Ang University, Seoul 156-756, Korea

* Author to whom Correspondence should be addressed; E-Mail: hansy@cau.ac.kr; Tel.: +82-10-2320-5327; Fax: +82-2-825-6996.

Received: 14 October 2009; in revised form: 18 November 2009 / Accepted: 1 December 2009 /

Published: 9 December 2009

Abstract: The paper proposes three alternative extensions to the classical global-best particle swarm optimization dynamics, and compares their relative performance with the standard particle swarm algorithm. The first extension, which readily follows from the well-known Lyapunov's stability theorem, provides a mathematical basis of the particle dynamics with a guaranteed convergence at an optimum. The inclusion of local and global attractors to this dynamics leads to faster convergence speed and better accuracy than the classical one. The second extension augments the velocity adaptation equation by a negative randomly weighted positional term of individual particle, while the third extension considers the negative positional term in place of the inertial term. Computer simulations further reveal that the last two extensions outperform both the classical and the first extension in terms of convergence speed and accuracy.

Keywords: particle swarm dynamics; metaheuristics; continuous function optimization; stability; convergence; lyapunov stability theorem

1. Introduction

The concept of particle swarms originated from the simulation of the social behavior commonly observed in animal kingdom and evolved into a very simple but efficient technique for global numerical optimization in recent past. The Particle Swarm Optimization (PSO) [1,2], as it is called now, does not require any gradient information of the function to be optimized, uses only primitive mathematical operators and is conceptually very simple. PSO emulates the swarming behavior of insects, animals herding, birds flocking and fish schooling, where these swarms forage for food in a collaborative manner. PSO also draws inspiration from the boids method of Craig Reynolds and Socio-Cognition [2].

Since its inception, the research on PSO has centered on the improvement of the particle dynamics and the algorithm. Shi and Eberhart incorporated the *inertia factor* [3] in the basic PSO dynamics for faster convergence of the algorithm. Clerc and Kennedy [4] considered in their work an alternative form of PSO dynamics using a parameter called *constriction factor*, and gave a detailed theoretical analysis to determine the value of the parameter. Eberhart and Shi compared the effect of inertia factor and constriction factor on PSO performance [5]. Angeline [6] introduced a form of selection operation in the PSO algorithm, so that the characteristics of good particles are transferred to the less effective members of the swarm to improve their behavior. Suganthan [7] employed a neighborhood operator in the basic particle swarm optimization scheme to study the swarm behavior. Extension of the PSO algorithm to deal with dynamic environment and efficient explorations are undertaken in [8,9]. Ratnaweera *et al.*, while proposing a new model of self-organizing hierarchical PSO [10], ignored the term involving inertia factor from the velocity adaptation rule. Another contribution of this paper is the inclusion of time-varying inertia weight and time-varying acceleration coefficients for better performance of the algorithm.

In [11], a new crossover operator is defined to swap information between two individuals in order to determine their next position on the search landscape. Miranda *et al.* in [12] proposed a mutation operator on the parameters of the PSO dynamics and the position of the neighborhood best particle, so as to enhance the diversity of the particles, thereby increasing the chances of escaping local minima. In [13], the inertia weight is mutated and the particles are relocated when they are too close to each other. A further increase in the diversity of the population has been attained in [14,15] through introduction of a new collision-avoiding mechanism among the particles. Xie *et al.* [16] added negative entropy to the PSO to discourage premature convergence. In [2], a cooperative PSO (CPSO) is implemented to significantly improve the performance of the classical PSO. Hendtlass *et al.* [17] combined Ant Colony Optimization with PSO to determine the neighborhood best of a particle from a list of best positions found so far by all the particles.

Most of existing works on PSO refer to single objective optimization problems. Coello *et al.* first proposed a formulation of multi-objective optimization problem using PSO [18], and later extended it to include a constraint-handling mechanism and a mutation operator [19] to improve the power of exploration of the optimization algorithm. Agrawal, Panigrahi and Tiwari [20] in a recent paper proposed a fuzzy clustering-based PSO algorithm to solve the highly constrained environmental/economic dispatch problem involving conflicting objectives.

There exists an extensive literature on improving the performance of the PSO algorithm. This has been undertaken by two alternative approaches. First, the researchers are keen to improve swarm

behavior by selecting the appropriate form of the swarm dynamics. Alternatively, considering a given form of particle dynamics, researchers experimentally, or theoretically, attempted to find the optimal settings of the range of parameters to improve PSO behavior. In this paper, we adopt the first policy to determine a suitable dynamics, and then attempted to empirically determine the optimal parameter settings.

The classical PSO dynamics adapts the velocity of individual particles by considering the inertia of the particle and the position of local and global attractors. The positions of the attractors are also adapted over the iterations of the algorithm. The motion of the particles thus continues until most of the particles converge in the close vicinity of the global optima. In this paper, we consider different versions of the swarm dynamics to study the relative performance of the PSO algorithm both from the point of view of accuracy and convergence time.

The formal basis of our study originates from the well-known Lyapunov's theorem of classical control theory. The Lyapunov's theorem is widely used in nonlinear system analysis to determine the necessary conditions for stability of a dynamical system. In this paper, we indirectly used Lyapunov's stability theorem to determine a dynamics that necessarily converges to an optima of the Lyapunov-like search landscape. The principles of guiding particle dynamics towards the global and local optima, here too, is ensured by adding local and global attractor terms to the modified PSO dynamics. The rationale of selecting a dynamics that converges at one of the optima on a multimodal surface, and the principle of forcing the dynamics to move towards local and global optima together makes it attractive for use in continuous nonlinear optimization.

There are, however, search landscapes that do not possess the necessary characteristics of a Lyapunov surface. This calls for an alternative dynamics, which maintains the motivation of this research but can avoid the restriction on the objective function to necessarily be Lyapunov-like. A look at the dynamics constructed for Lyapunov-like benchmark functions essentially reveals an inclusion of a negative position term in the velocity adaptation rule. This prompted us to realize different variants of the classical PSO dynamics, such as (a) replacement of the inertial term by a negative partial derivative of the Lyapunov-like search landscape, (b) inclusion of a negative particle position in the velocity adaptation rule, (c) replacement of the inertial term by the negative positional term in the dynamics. Computer simulations undertaken on a set of eight benchmark functions reveals that the modifications in the PSO dynamics results in a significant improvement in the PSO algorithm with respect to both convergence speed and accuracy. Note that the extensions developed in this article are primarily meant for fast and accurate optimization of continuous and differentiable functions, as all of them involve first derivatives of the objective function to be used.

The rest of the paper is organized as follows. Section 2 provides a set of formal definitions on Lyapunov stability of nonlinear dynamics. It explains the basis of selection of a dynamics for a given Lyapunov-like objective function. The rationale of speed-up of the proposed swarm algorithm using the selected dynamics is given in this section. Experimental results over several numerical benchmarks are presented in Section 3. Finally the paper is concluded in Section 4.

2. Proposed Extensions of the Classical PSO Dynamics

In this section, we briefly outline one typical PSO dynamics, and the PSO algorithm. We next present the possible modifications that we need to undertake in the dynamics to study their relative performance with the classical PSO algorithm.

The global-best (g-best) PSO dynamics for the j^{th} particle is given in vector form through Equations 1 and 2:

$$\mathbf{V}_j(t+1) = \omega \mathbf{V}_j(t) + \alpha^l(t)(\mathbf{P}_j^l(t) - \mathbf{X}_j(t)) + \alpha^g(t)(\mathbf{P}^g(t) - \mathbf{X}_j(t)) \quad (1)$$

$$\mathbf{X}_j(t+1) = \mathbf{X}_j(t) + \mathbf{V}_j(t+1) \quad (2)$$

where:

$\mathbf{V}_j(t) = [v_{j1}(t) \ v_{j2}(t) \dots \dots \dots v_{jD}(t)]$ is a D -dimensional velocity vector at time t ,

$\mathbf{X}_j(t) = [x_{j1}(t) \ x_{j2}(t) \dots \dots \dots x_{jD}(t)]$ is a D -dimensional position vector at time t ,

$\mathbf{P}_j^l(t) = [p_{j1}^l(t) \ p_{j2}^l(t) \dots \dots \dots p_{jD}^l(t)]$ is a D -dimensional personal (local) best position vector of particle j , so far achieved until iteration t ,

$\mathbf{P}_j^g(t) = [p_{j1}^g(t) \ p_{j2}^g(t) \dots \dots \dots p_{jD}^g(t)]$ is a D -dimensional global best position vector found so far by the entire swarm at iteration t .

Empirically, ω is a random no. in $[0,1]$, $\alpha^l(t)$ and $\alpha^g(t)$ are random coefficients in $[0, 2]$ and $[0, 4]$ respectively. Inertia factor ω is selected randomly only once in the PSO algorithm, whereas $\alpha^l(t)$ and $\alpha^g(t)$ are selected randomly in each iteration of the PSO algorithm.

The basic PSO algorithm is presented here for convenience of the readers. The notion of time t is dropped from the algorithm for simplicity.

PSO-Algorithm

Begin

Initialize population;

While terminating conditions not reached **do**

Begin

For $j = 1$ to N **do**

// $N =$ Number of particles//

Begin

Evaluate fitness $f(\bullet)$ of particle j ;

If $f(\mathbf{X}_j) < f(\mathbf{P}_j^l)$

$\mathbf{P}_j^l \leftarrow \mathbf{X}_j$;

End for;

$\mathbf{P}^g \leftarrow \text{Arg}[\text{Min}\{f(\mathbf{P}_1^l), f(\mathbf{P}_2^l), \dots, f(\mathbf{P}_N^l)\}]$;

For $j = 1$ to N **do**

Begin

Adapt $\mathbf{V}_j \leftarrow \omega \mathbf{V}_j + \alpha^l(\mathbf{P}_j^l - \mathbf{X}_j) + \alpha^g(\mathbf{P}^g - \mathbf{X}_j)$;

Adapt $\mathbf{X}_j = \mathbf{X}_j + \mathbf{V}_j$;

End for;

End while;

End.

A look at the PSO algorithm reveals that it attempts to determine the optima on a search landscape by allowing several particles (agents) to explore on the surface with an ultimate aim to terminate at the global optima. The terminating condition usually includes an upper limit on the iterations or a lower limit to the unsigned successive difference in the best particle position, or whichever occurs earlier.

In the next sub-section, we would look for a dynamics that has a tendency to move towards optima, which need not essentially be the global optima. This can be attained by identifying a suitable dynamics that ensures asymptotic stability in the vicinity of an optimum over the search landscape. This, of course, needs additional restriction on the surface to satisfy the necessary conditions to be Lyapunov-like [24]. If a suitable dynamics ensuring the convergence to an optimum is identified, we can control the motion of the particles towards the global/local optima by adding global and local attractors in the dynamics as used in the PSO dynamics.

2.1. Identifying a Stable Dynamics for a Lyapunov-like Surface

This section begins with a few definitions, available in the standard literature in Nonlinear Control Theory in [21-24], to explain the methodology of determining a stable dynamics for a Lyapunov-like surface. In what follows we shall use the following special notations: $\|\mathbf{X}\|$ to define the Euclidean norm of a vector.

$S(\varepsilon)$ to denote an ε -neighborhood surrounding a point defined by the position-vector \mathbf{X}_e . $S(\varepsilon)$ is basically a set containing all the points in the vector space for which $\|\mathbf{X} - \mathbf{X}_e\| \leq \varepsilon$.

Definition 2.1. A point $\mathbf{X} = \mathbf{X}_e$ is called an *equilibrium state*, if the dynamics of the system which is given by

$$\frac{d\mathbf{X}}{dt} = f(\mathbf{X}(t))$$

becomes zero at $\mathbf{X} = \mathbf{X}_e$ for any t . The equilibrium state is also called equilibrium (stable) point in D -dimensional hyperspace, when the state \mathbf{X}_e has D -components.

Definition 2.2. A scalar function $V(\mathbf{X})$ is said to be *positive definite* with respect to the point \mathbf{X}_e in the region $\|\mathbf{X} - \mathbf{X}_e\| \leq K$, if $V(\mathbf{X}) > 0$ at all points of the region except at \mathbf{X}_e where it is zero.

Note that $V(\mathbf{X})$ will be called *negative definite* if $-V(\mathbf{X})$ is positive definite. A scalar function $V(\mathbf{X})$ is said to be *indefinite* in the region $\|\mathbf{X} - \mathbf{X}_e\| \leq K$, if it assumes both positive and negative values within this region.

Definition 2.3. A scalar function $V(\mathbf{X})$ is said to be *positive semi-definite* with respect to the point \mathbf{X}_e in the region $\|\mathbf{X} - \mathbf{X}_e\| \leq K$, if its value is positive at all points of the region except at finite number of points including origin where it is zero.

Note that similar as definition 2.2, the scalar function $V(\mathbf{X})$ is said to be *negative semi-definite* if $-V(\mathbf{X})$ is positive semi-definite.

Definition 2.4. A scalar function $V(\mathbf{X})$ is called a *Lyapunov surface* with respect to the origin, if it satisfies the three conditions listed below:

- i. $V(\mathbf{0}) = 0$
- ii. $V(\mathbf{X}) > 0$ for $\mathbf{X} \neq \mathbf{0}$
- iii. $\frac{\partial V}{\partial x_i}$ is a continuous function of x_i , where x_i is the i^{th} component of \mathbf{X}

Definition 2.5. A dynamics $d\mathbf{X}/dt = f(\mathbf{X}(t))$ is *asymptotically stable* at the equilibrium point \mathbf{X}_e , if

- (a) it is stable in the sense of Lyapunov, *i.e.*, for any neighborhood $S(\varepsilon)$ surrounding \mathbf{X}_e there is a region $S(\delta)$, $\delta < \varepsilon$, such that trajectories of the dynamics starting within $S(\delta)$ do not leave $S(\varepsilon)$ as time $t \rightarrow \infty$ and
- (b) the trajectory starting within $S(\delta)$ converges to the origin as time t approaches infinity

The sufficient condition for stability of a dynamics can be obtained from the Lyapunov's theorem, presented below.

Lyapunov's Stability Theorem [21]: Given a scalar function $V(\mathbf{X})$ and some real number $\varepsilon > 0$, such that for all \mathbf{X} in the region $S(\varepsilon)$ the following conditions hold:

- 1) $V(\mathbf{X}_e) = 0$
- 2) $V(\mathbf{X})$ is positive definite.
- 3) $V(\mathbf{X})$ has continuous first partial derivatives with respect to all components of \mathbf{X}

Then the equilibrium state \mathbf{X}_e of the system $d\mathbf{X}/dt = f(\mathbf{X}(t))$ is

- a) *asymptotically stable* if dV/dt is negative definite, and
- b) *asymptotically stable* in the large if dV/dt is negative definite, and in addition, $V(\mathbf{X}) \rightarrow \infty$ as $\|\mathbf{X} - \mathbf{X}_e\| \rightarrow \infty$

Example 1

Let $V(\mathbf{X}) = x_1^2 + x_2^2$ be a Lyapunov energy function for the given dynamics $\frac{dx_1}{dt} = -x_1$ and $\frac{dx_2}{dt} = -x_2$ with the equilibrium point $\mathbf{X}_e = [0,0]$. Then:

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial x_1} \frac{dx_1}{dt} + \frac{\partial V}{\partial x_2} \frac{dx_2}{dt} \\ &= 2x_1(-x_1) + 2x_2(-x_2) \\ &= -2(x_1^2 + x_2^2) < 0 \end{aligned}$$

i.e., negative definite.

Here, $V(\mathbf{X})$ satisfies the first two criterions indicated in the theorem, and the partial derivatives $\partial V/\partial x_1$ and $\partial V/\partial x_2$ are also continuous functions of x_1 and x_2 . Consequently, the asymptotic stability of

the dynamics is ensured as dV/dt is found to be negative definite for all points except at $x = 0$. Further, as $\|\mathbf{X}\| \rightarrow \infty$, $V(\mathbf{X})$ also approaches infinity. Therefore, the asymptotic stability of the dynamics in the large is also ascertained.

The condition for asymptotic stability, as indicated in Theorem1, can be applied to the particle swarm optimization to ensure stability of the dynamics, thereby reducing the convergence time of the algorithm.

When all the three underlying conditions of a Lyapunov function, indicated in Definition 2.4 are supported by the objective function, we would be interested to determine the dynamics that satisfies the necessary conditions for asymptotic stability of the dynamics. It follows from Lyapunov's Theorem that the asymptotic stability of an equilibrium state guided by the dynamics dx_i/dt is ascertained if:

$$\frac{df}{dt} = \sum_{i=1}^D \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} < 0 \quad (3)$$

The inequality (3) essentially holds when:

$$\frac{dx_i}{dt} = - \frac{\partial f}{\partial x_i} \quad (4)$$

It is indeed important to note that the condition (4) holds for the i -th dimension of a particle roaming over the Lyapunov-like surface for $1 \leq i \leq D$.

Example 2

In this example, we would like to determine a stable dynamics for a Lyapunov-like objective function. Consider for instance the Griewank function in D-dimension:

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

In order to have asymptotic stability of the dynamics, we set:

$$\frac{dx_i}{dt} = - \frac{\partial f}{\partial x_i} = \frac{x_i}{2000} + \frac{1}{\sqrt{i}} \sin\left(\frac{x_i}{\sqrt{i}}\right) \left[\prod_{j=1, j \neq i}^D \cos\left(\frac{x_j}{\sqrt{j}}\right) \right]$$

It is also apparent to note that the given function $f(\mathbf{X})$ satisfies the three necessary conditions of a Lyapunov function. Now, if we replace the term involving inertia factor by the obtained value of dx_i/dt in the PSO dynamics, then the PSO is expected to converge very quickly as the necessary condition for asymptotic stability has been satisfied while deriving the dynamics:

$$\frac{dx_i}{dt} = \frac{x_i}{2000} + \frac{1}{\sqrt{i}} \sin\left(\frac{x_i}{\sqrt{i}}\right) \left[\prod_{j=1, j \neq i}^D \cos\left(\frac{x_j}{\sqrt{j}}\right) \right]$$

Table 1 provides a list of eight typical benchmark functions along with the derived expressions for dx_i/dt that ensures the asymptotic stability of the derived dynamics over the Lyapunov-like objective function.

We now define Lyapunov-based PSO dynamics (LyPSO) by adding the local and global attractor terms of classical PSO to the derived expression for asymptotically stable Lyapunov dynamics, given in Equations (5) and (6):

$$\mathbf{V}_j(t+1) = -\omega \cdot \vec{\nabla} f(\mathbf{X}) + \alpha^l(t)(\mathbf{P}'_j(t) - \mathbf{X}_j(t)) + \alpha^g(t)(\mathbf{P}^g_j(t) - \mathbf{X}_j(t)) \tag{5}$$

$$\mathbf{X}_j(t+1) = \mathbf{X}_j(t) + \mathbf{V}_j(t+1) \tag{6}$$

Note that $\vec{\nabla} f = [\partial f / \partial x_1, \dots, \partial f / \partial x_D]$ denotes the gradient of the scalar function f to be optimized. The first term in the right hand side of Equation (5) ensures motion of the particle towards minima, while the second and third term controls the motion towards local and global optima respectively. It is apparent from Table 1 that dx_i/dt obtained for different Lyapunov-like surfaces include a factor of $(-x_i)$. The condition for $dx_i/dt = -\omega x_i$ is tabulated for all the eight benchmark functions in Table 2. Consequently, instead of computing dx_i/dt by the approach stated earlier, we can simply add a term $-\omega x_i$ to the i^{th} component of the updated velocity in the classical PSO. The resulting dynamics then looks like Equations (7) and (8):

$$\mathbf{V}_j(t+1) = -\omega \cdot \mathbf{X}_{j,i}(t) + \omega \cdot \mathbf{V}_{j,i}(t) + \alpha^l(t)(\mathbf{P}'_j(t) - \mathbf{X}_j(t)) + \alpha^g(t)(\mathbf{P}^g_j(t) - \mathbf{X}_j(t)) \tag{7}$$

$$\mathbf{X}_j(t+1) = \mathbf{X}_j(t) + \mathbf{V}_j(t+1) \tag{8}$$

The dynamics given by Equations (7) and (8) is referred to as Position-based PSO (PPSO).

Table 1. The derived dynamics for the selected benchmark functions.

Function name	Functional form $f(\mathbf{X}(t))$	dx_i / dt
Sphere Function	$f(\mathbf{X}) = \sum_{i=1}^D x_i^2$	$-2x_i$
Rosenbrock's Function	$f(\mathbf{X}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-(400x_i^2 + 2)x_i + 2(1 + 200x_i x_{i+1})$
Step Function	$f(\mathbf{X}) = \sum_{i=1}^D (x_i + 0.5)^2$	$-2 x_i + 0.5 $
Schwefel's Problem 1.2	$f(\mathbf{X}) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$-2x_i \left(\sum_{j=1}^{i-1} x_j \right)$
Rastrigin's Function	$f(\mathbf{X}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$-2x_i - 20\pi \sin(2\pi x_i)$
Ackley's Function	$f(\mathbf{X}) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i) + 20 + e$	$-\frac{4}{D} \left(\exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) \right) \frac{x_i}{\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}} - \frac{2\pi}{D} \sin 2\pi x_i \left[\exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) \right]$
Griewank's Function	$f(\mathbf{X}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-\frac{x_i}{2000} - \frac{1}{\sqrt{i}} \sin\left(\frac{x_i}{\sqrt{i}}\right) \left[\prod_{j=1, j \neq i}^D \cos\left(\frac{x_j}{\sqrt{j}}\right) \right]$
Salomon's function	$f(\mathbf{X}) = -\cos\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2} + 1$	$-2\pi x_i \left(\sum_{i=1}^D x_i^2\right)^{-\frac{3}{2}} \sin\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) - 0.1 x_i \left(\sum_{i=1}^D x_i^2 + 1\right)^{-\frac{3}{2}}$

Table 2. Reduced form of dx_i/dt .

Function name	Reduced form of $\frac{dx_i}{dt}$	Condition for reduction
Sphere Function	$-2x_i$	Unconditional
Rosenbrock's Function	$-x_i(400x_i^2 + 2 - 400x_{i+1})$	When $x_i x_{i+1} \gg -\frac{1}{200}$
Step Function	$-2 x_i $	When $x_i \gg 0.5$
Schwefel's Problem 1.2	$-2x_i \left(\sum_{j=1}^{i-1} x_j \right)$	Unconditional
Rastrigin's Function	$-x_i(2 + 4\pi^2)$	When x_i is very small.
Ackley's Function	$-x_i \left(\frac{4}{D} \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) + \frac{2\pi}{D} \left[\exp \left(\frac{1}{D} \right) \right] \right) \frac{x_i}{\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}}$	When x_i is very small.
Griewank's Function	$-x_i \left(\frac{1}{2000} + \frac{1}{i} \right)$	When x_i is very small
Salomon's function	$-x_i \left(2\pi \left(\sum_{i=1}^D x_i^2 \right)^{-\frac{3}{2}} \sin \left(2\pi \sqrt{\sum_{i=1}^D x_i^2} \right) + 0.1 \left(\sum_{i=1}^D x_i^2 + 1 \right)^{-\frac{3}{2}} \right)$	Unconditional

For the sake of completeness of our study, we consider a third category of the dynamics, where the inertial term is dropped from the PSO dynamics, indicated in Equations (9) and (10). The modified dynamics, called Steepest-PSO (SPSO) for its fast convergence (vide Section 3), is formally given below:

$$\mathbf{V}_j(t+1) = -\omega \mathbf{X}_j(t) + \alpha^l(t)(\mathbf{P}_j^l(t) - \mathbf{X}_j(t)) + \alpha^g(t)(\mathbf{P}_j^g(t) - \mathbf{X}_j(t)) \quad (9)$$

$$\mathbf{X}_j(t+1) = \mathbf{X}_j(t) + \mathbf{V}_j(t+1) \quad (10)$$

In the next section, we would justify the reason for accuracy and speed-up of PPSO and SPSO over classical PSO.

2.2. The Rationale of Speed-up of the PPSO and SPSO Dynamics over Classical PSO

To compare the relative performance in speed-up and convergence of the proposed algorithms, we study the stability behavior of the proposed PPSO and SPSO dynamics, in absence of the local and the

global attractors. This is performed by solving the first order difference equations. The condition for asymptotic stability and the location of the stable point can be ascertained from the solution of the dynamics. Theorems 1 to 2 provide interesting results, indicating asymptotic stability of the SPSO and PPSO dynamics to the origin irrespective of the search landscape, whereas Theorem 3 indicates asymptotic stability of the classical PSO to a stable point, which need not essentially be the origin. The rate at which the particle position approaches the origin further indicates that the speed of convergence of the SPSO algorithm far exceeds that of PPSO, while the speed of PPSO algorithms beats classical PSO.

Theorem 1: *The dynamics of the j^{th} particle in the i^{th} dimension given by*

$$v_{j,i}(t+1) = -\omega x_{j,i}(t) \quad (11)$$

has a stable point at the origin, when $\omega \leq 1$.

Proof. Let E be an extended difference operator, such that

$$E(x_{j,i}(t)) = x_{j,i}(t) + \Delta x_{j,i}(t) = (1 + \Delta)x_{j,i}(t) = x_{j,i}(t+1)$$

Now extending the concept of derivatives to the discrete time domain, Equation (11) now can be written as

$$\begin{aligned} \frac{x_{j,i}(t+1) - x_{j,i}(t)}{(t+1) - (t)} &= -\omega x_{j,i}(t) \\ \Rightarrow x_{j,i}(t+1) - x_{j,i}(t) &= -\omega x_{j,i}(t) \end{aligned} \quad (12)$$

Replacing $x_{j,i}(t+1)$ by $E(x_{j,i}(t))$ in (12) we obtain:

$$\begin{aligned} E x_{j,i}(t) - (1 - \omega)x_{j,i}(t) &= 0 \\ \Rightarrow (E - 1 + \omega)x_{j,i}(t) &= 0 \\ \therefore E &= 1 - \omega \end{aligned}$$

Consequently, the solution of the dynamics (11) is given by

$$x_{j,i}(t) = A(1 - \omega)^t \quad (13)$$

where A is a constant. The expression (13) indicates that for $\omega < 1$, $x_{j,i}(t) \rightarrow 0$ when $t \rightarrow \infty$. Therefore, the dynamics is asymptotically stable at the origin for $\omega < 1$. When $\omega = 1$, $x_{j,i}(t) = 0$ at all time t . Hence, the theorem follows.

Theorem 2: *The dynamics of the j^{th} particle in the i^{th} dimension given by*

$$v_{j,i}(t+1) = \omega v_{j,i}(t) - \omega x_{j,i}(t) \quad (14)$$

Is asymptotically stable with a stable point at the origin, when $\omega < 1$.

Proof. We can rewrite Equation (14) as

$$\frac{x_{j,i}(t+1) - x_{j,i}(t)}{(t+1) - (t)} = -\omega x_{j,i}(t) + \omega x_{j,i}(t) - \omega x_{j,i}(t-1) \quad (15)$$

Replacing $x_{j,i}(t+1)$ by $E(x_{j,i}(t))$ and $x_{j,i}(t-1)$ by $E^{-1}(x_{j,i}(t))$ in Equation (15), we obtain

$$\begin{aligned} &\Rightarrow Ex_{j,i}(t) - x_{j,i}(t) + \omega E^{-1}x_{j,i}(t) = 0 \\ &\Rightarrow (E^2 - 1 + \omega)x_{j,i}(t) = 0 \\ &\Rightarrow E^2 = 1 - \omega \\ &\Rightarrow E = \pm\sqrt{1 - \omega} \end{aligned}$$

So, the solution of the dynamics (16) is given by:

$$x_{j,i}(t) = A(\sqrt{1 - \omega})^t - B(\sqrt{1 - \omega})^t \quad (16)$$

where A and B are constants. It is apparent from expression (16) that $x_{j,i}(t)$ asymptotically converges to the origin for $\omega < 1$. Therefore, the dynamics is asymptotically stable with a stable point at the origin for $\omega < 1$. When $\omega = 1$, $x_i(t) = 0$ for all t . This proves the Theorem.

Theorem 3: *The dynamics of j^{th} particle in the i^{th} dimension given by:*

$$v_{j,i}(t+1) = \omega v_{j,i}(t) \quad (17)$$

is asymptotically stable and it converges to a stable point, which need not essentially be zero.

Proof. We can rewrite Equation (17) as:

$$\frac{x_{j,i}(t+1) - x_{j,i}(t)}{(t+1) - t} = \omega \left[\frac{x_{j,i}(t) - x_{j,i}(t-1)}{t - (t-1)} \right] \quad (18)$$

Let E be an extended difference operator, such that $E^n x_{j,i}(t) = x_{j,i}(t+n)$, and $E^{-n} x_{j,i}(t) = x_{j,i}(t-n)$ for any positive integer n . Consequently, Equation (18) is transformed to:

$$\begin{aligned} &Ex_{j,i}(t) - (1 + \omega)x_{j,i}(t) + \omega E^{-1}x_{j,i}(t) = 0, \\ &\Rightarrow E^2 x_{j,i}(t) - (1 + \omega)Ex_{j,i}(t) + \omega x_{j,i}(t) = 0, \\ &\Rightarrow (E^2 - (1 + \omega)E + \omega)x_{j,i}(t) = 0, \\ &\Rightarrow (E^2 - E - \omega E + \omega) = 0, \\ &\Rightarrow (E - \omega)(E - 1) = 0. \\ &\therefore E = \omega, 1. \end{aligned}$$

So, the solution of the dynamics (17) is given by:

$$\begin{aligned} x_{j,i}(t) &= A(1)^t + B\omega^t \\ &\Rightarrow x_{j,i}(t) = A + B\omega^t, \end{aligned} \quad (19)$$

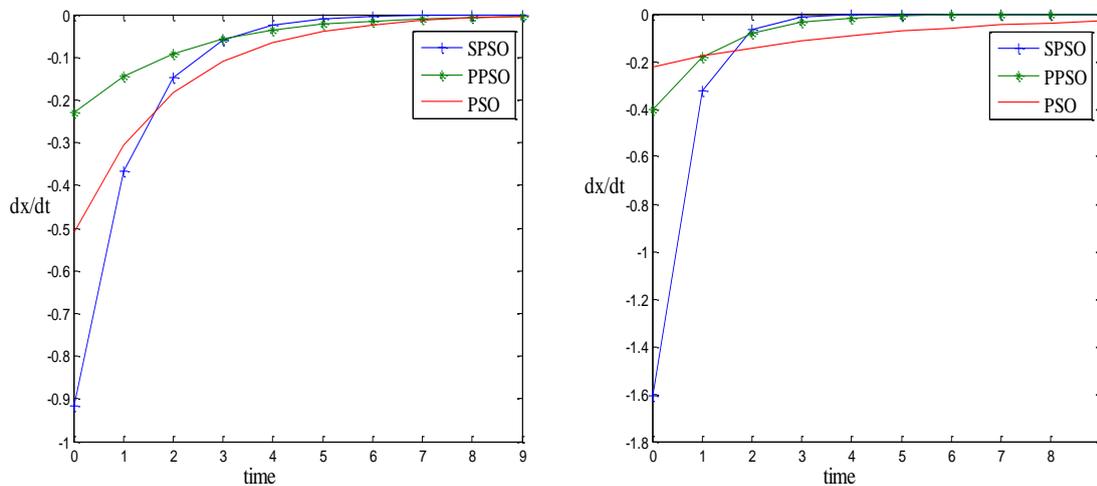
where A and B are constants. It is apparent from expression (21) that $x_{j,i}(t)$ asymptotically converges to A as time t approaches infinity. Since A is not zero unconditionally, therefore the statement of the theorem follows.

Table 3 provides the results of computation of dx_i/dt for SPSO, PPSO and classical PSO. The computations in the table are performed from Equations 13, 16 and 19 respectively. Figure 1 shows the variation of dx_i/dt with respect to time for $\omega = 0.6$ and $\omega = 0.8$. It is apparent from the graphs that in the absence of local and global attractors, the dynamics of SPSO converges faster than that of PPSO, which further converges faster than the classical PSO.

Table 3. dx_i/dt for SPSO, PPSO and classical PSO.

Dynamics	dx_i/dt
SPSO	$A(1-\omega)^t \log_e(1-\omega)$
PPSO	$\frac{(A-B)}{2}(1-\omega)^{t/2} \log_e(\sqrt{1-\omega})$
Classical PSO	$B\omega^t \log_e \omega$

Figure 1. Variation of dx_i/dt with respect to time: (a) for $\omega = 0.6$. (b) for $\omega = 0.8$.



3. Computer Simulations and Experimental Results

3.1. Benchmarks

In order to study the performance of the proposed three alternative PSO dynamics, we used eight well-known benchmark functions as listed in Table 1. All the functions listed here have global minima at the origin except the Rosenbrock function. The performance of the three proposed dynamics for these eight functions is compared with that of classical PSO.

3.2. Parametric Range and Error Criterion

Early methods of performance evaluations for evolutionary algorithms were restricted to symmetric initializations. In recent time, researchers prefer asymmetric initialization [25]. We here used the asymmetric initialization method to evaluate the performance of proposed three dynamics along with the classical PSO. In Table 4, we provide the initialization range of the objective function variables, the position of theoretical optima and the error criterion used to terminate the algorithm. Different error criteria were used for different benchmark functions.

Table 4. Parametric range of benchmark functions.

Function Name	Dimension	Initialization Range	Theoretical Optima	Error Criterion
Sphere Function	30	[50, 100]	[0,0.....,0]	0.01
Rosenbrock's Function	30	[15, 30]	[1,1.....,1]	0.001
Step Function	30	[50, 100]	[0,0.....,0]	0.01
Schwefel's Problem 1.2	30	[50, 100]	[0,0.....,0]	0.001
Rastrigin's Function	30	[2.56, 5.12]	[0,0.....,0]	0.1
Ackley's Function	30	[15, 32]	[0,0.....,0]	0.01
Griewank's Function	30	[300, 600]	[0,0.....,0]	0.001
Salomon's function	30	[50, 100]	[0,0.....,0]	0.001

3.3. Simulation Strategies

Parameter selection of the PSO dynamics also is a crucial issue for speed-up and accuracy of the PSO algorithm. For a given benchmark function, we initially took wider range of the PSO dynamics parameters: $\alpha^g(t)$, $\alpha^l(t)$ and ω . The initial ranges selected in our simulation were $\alpha^g(t)$ in [0, 4], $\alpha^l(t)$ in [0, 2] and $\omega < 1$. Several hundred runs of the PSO programs with random parameter settings in the above ranges confirm that for a specific function, the best choice of parameters are restrictive as indicated in Table 5.

Table 5. Range of optimal values of $\alpha^g(t)$, $\alpha^l(t)$ and ω of LyPSO, PPSO and SPSO.

Function Name	Parameters for PSO Algorithm					
	$\alpha^g(t)$, $\alpha^l(t)$ and ω of LyPSO			$\alpha^g(t)$, $\alpha^l(t)$ and ω of PPSO/SPSO		
	$\alpha^g(t)$	$\alpha^l(t)$	ω	$\alpha^g(t)$	$\alpha^l(t)$	ω
Sphere Function	0.9999–1.9999	0.0001–0.001	0.5–0.7	0.1999–1.9	0.0001–0.01	0.4–0.7
Rosenbrock's Function	0.1999–0.3999	0.001–0.003	10^{-12} – 10^{-11}	0.001–0.999	0.001–0.009	0.3–0.6
Step Function	0.9999–1.9999	0.0001–0.001	0.5–0.7	0.199–0.999	0.001–0.01	0.3–0.7
Schwefel's Problem 1.2	0.599–0.799	0.001–0.003	10^{-12} – 10^{-9}	0.199–0.999	0.001–0.01	0.4–0.7
Rastrigin's Function	0.2999–0.5999	0.0001–0.0005	0.0001–0.0009	0.2999–0.5999	0.0001–0.0005	0.3–0.6
Ackley's Function	0.1–0.2	0.7–0.8	0.001	Random	Random	0.3–0.7
Griewank's Function	Random	Random	56–60	Random	Random	0.3–0.7
Salomon's function	0.1999–0.5999	0.0001–0.0005	5000	0.1999–0.5999	0.0001–0.0005	0.3–0.6

The following observations readily follow from Table 5.

Observation 1: For each benchmark function, the parameter set of the dynamics including $\alpha^g(t)$, $\alpha^l(t)$ and ω for LyPSO has a relatively restricted range than those of PPSO and SPSO.

Observation 2: The parameter sets for most of the benchmark functions for the LyPSO dynamics have a common range as listed below: $\alpha^g(t)$ in [0.1999, 0.5999] and $\alpha^l(t)$ in [0.0001, 0.001]. The parameter

sets for most of the benchmark functions for the PPSO and SPSO dynamics have a common range as listed below: $\alpha^s(t)$ in $[0.199, 0.999]$, $\alpha^l(t)$ in $[0.0001, 0.01]$ and ω in $[0.3, 0.6]$. Moreover, the size of the population is taken as 40 and a maximum of 5,000 iterations were taken for 30-dimensional particles.

3.4. Experimental Results from the Simulations

The relative comparison of the convergence time of the three algorithms with respect to classical PSO are given in Figure 2a–h. It is observed from these figures that SPSO always outperforms PPSO in convergence time and accuracy. It is further revealed from these graphs that PPSO yields better performance in accuracy and convergence time with respect to both classical PSO and LyPSO. The performance of the four algorithms is summarized with a ‘ \leq ’ operator, where, $x \leq y$ indicates that performance of y is better than or equal to that of x. Relative performance:

$$\text{Classical PSO} \leq \text{LyPSO} \leq \text{PPSO} \leq \text{SPSO}$$

Figure 2. Progress towards the optima: (a) Sphere function. (b) Rosenbrock’s function. (c) Step function. (d) Schwefel’s Problem 1.2. (e) Rastrigin’s function. (f) Ackley’s function. (g) Griewank’s function. (h) Salomon’s function.

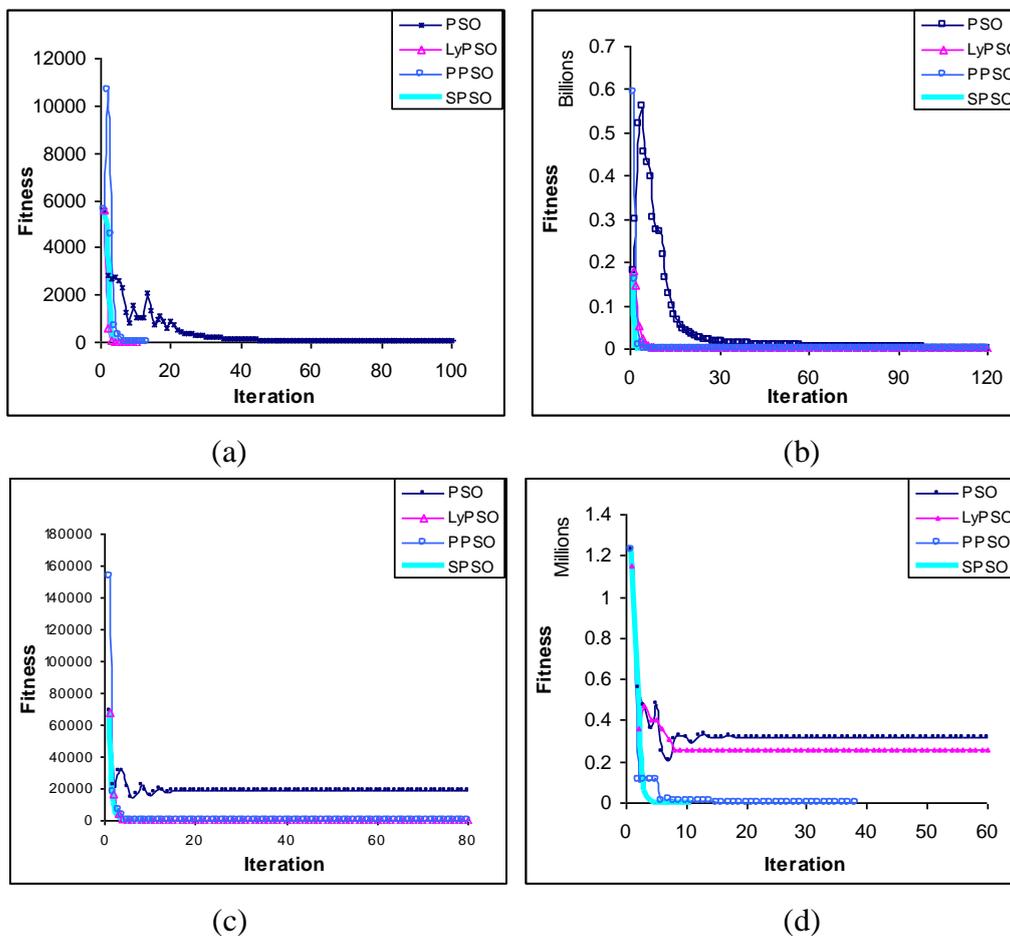


Figure 2. Cont.

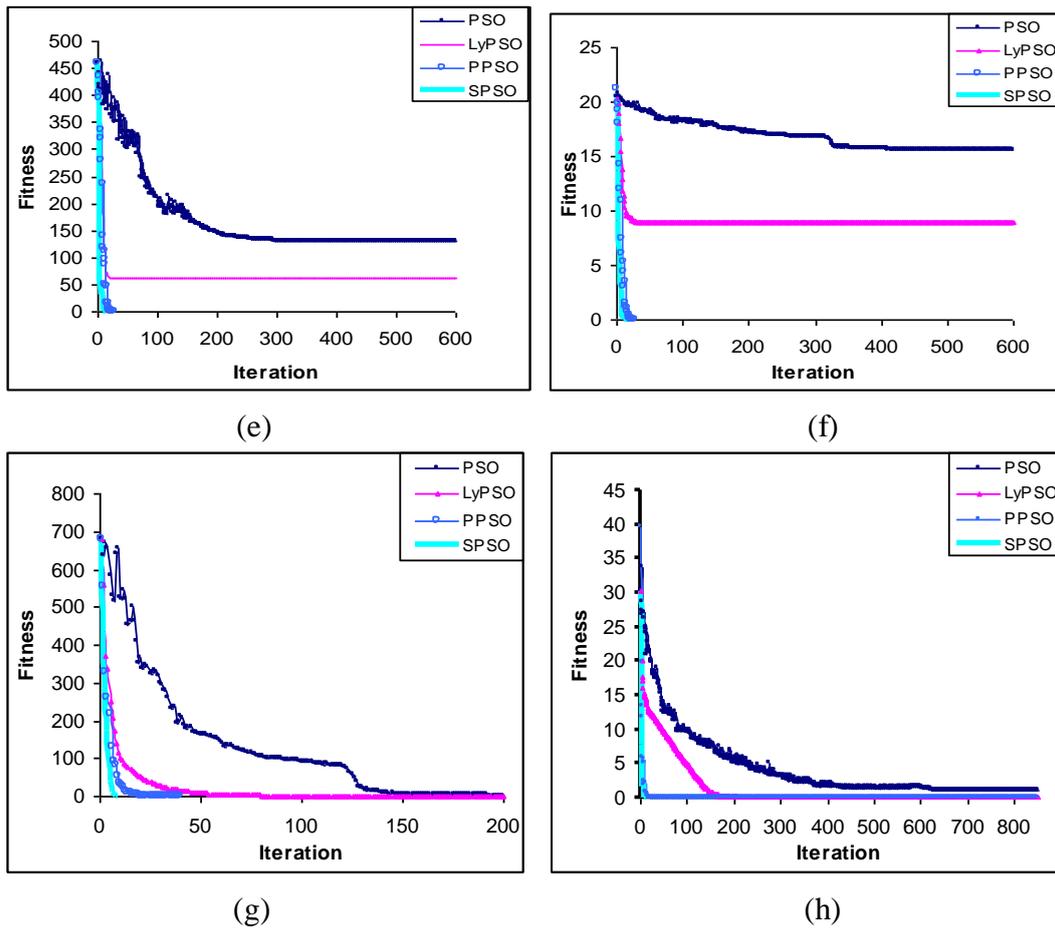


Table 6 provides the mean error and standard deviation for the globally best particle obtained by execution of the PPSO, SPSO, LyPSO and classical PSO over eight benchmark functions. The error was obtained by taking the Euclidean distance between the theoretical optima and the position of the best-fit particle for a given program run. The mean error designates the average of errors over 50 independent runs. In order to make the comparison fair enough, runs of all the algorithms were let start from the same initial population. The variance denotes the second moment of the errors with respect to the mean error. It is clear from Table 6 that for mean error for the SPSO algorithm is comparable but less than that obtained by PPSO algorithm, and the mean error obtained by the PPSO algorithm is insignificantly less than that of LyPSO algorithm. Further, the mean error obtained by the LyPSO algorithm is less in comparison to that of the classical PSO algorithm. This confirms that the SPSO algorithm outperforms the PPSO and LyPSO and definitely the classical PSO algorithm from the point of view of accuracy in solution. The speed-up of the SPSO algorithm has already been demonstrated in graphs *vide* Figures 2a–2h.

Table 7 shows results of unpaired *t*-tests between the best and second best algorithms in each case (standard error of difference of the two means, 95% confidence interval of this difference, the *t* value, and the two-tailed *P* value). For all cases, sample size = 50 and degrees of freedom = 98. It is interesting to see from Tables 6 and 7 that one or more of the proposed PSO methods can always beat the classical PSO in a statistically significant way.

Table 6. Mean error and standard deviation over the benchmarks.

Function Name	Dimension	Classical PSO	LyPSO	PPSO	SPSO
		Mean Error (Standard Deviation)			
Sphere Function	30	2.04e+00 (1.08e+00)	4.3e−03 (7.94e−04)	1.32e−02 (4.00e−03)	4.3e−03 (7.94e−04)
Rosenbrock's Function	30	7.77e+00 (0.77e+00)	1.58e+00 (2.07e−01)	9.99e−01 (9.4e−04)	9.94e−01 (2.7e−03)
Step Function	30	2.45e+01 (1.56e+00)	5.00e−01 (1.00e−03)	3.42e−01 (8.62e−02)	2.46 e−01 (1.01e−01)
Schwefel's Problem 1.2	30	4.2.4e+01 (6.35e+00)	1.89e+01 (1.86e+00)	2.7e−03 (1.3e−03)	1.90e−03 (1.20e−03)
Rastrigin's Function	30	2.97e+00 (1.9e−01)	1.48e+00 (8.06e−02)	2.70e−03 (8.27e−04)	2.79e−04 (6.82e−05)
Ackley's Function	30	7.03e+00 (6.22e+00)	2.76e+00 (3.85e−01)	1.70e−03 (6.20e−04)	1.74e−04 (4.95e−05)
Griewank's Function	30	1.73e+00 (1.13e+00)	9.93e−01 (3.00e−03)	5.17e−02 (1.81e−02)	1.58e−02 (3.50e−03)
Salomon's function	30	1.14e+01 (9.66e+00)	4.44e+00 (1.25e+00)	2.43e−01 (1.57e−01)	6.03e−04 (1.65e−04)

Table 7. Results of unpaired t-tests on the data of Table 6.

Function	Std. Err	<i>t</i>	95% Conf. Intvl	Two-tailed <i>P</i>	Significance
Sphere Function	0.000	21040.9635	−0.16340 to −0.16337	<0.0001	Extremely significant
Rosenbrock's Function	0.000	13.3484	−0.00585820 to −0.00434179	<0.0001	Extremely significant
Step Function	0.019	5.1050	−0.1329012 to −0.0584988	<0.0001	Extremely significant
Schwefel's Problem 1.2	0.000	3.1974	−0.0012965 to −0.0003035	0.0019	Very statistically significant
Rastrigin's Function	0.000	206.2488	−0.002444193 to −0.002397606	<0.0001	Extremely Significant
Ackley's Function	0.000	191.1514	0.0016651516 to 0.0017000883	<0.0001	Extremely Significant
Griewank's Function	0.007	4.8989	−0.0504426 to −0.0213574	<0.0001	Extremely significant
Salomon's function	0.022	10.9511	−0.286844994 to −0.198834365	<0.0001	Extremely significant

Our experimental results suggest that for multi-modal problems having the fitness landscape punctuated with multiple local optima, the SPSO dynamics is the most preferable choice. However, for uni-modal functions, LyPSO and SPSO are nearly equivalent in terms of their final accuracy and convergence speed.

In order to compare the scalability of the proposed PSO-variants against the growth of dimensionality of the search space, we need to plot the no. of fitness function evaluations with dimension of the search landscape. The results shown in figures are average over 50 independent runs of the PSO program. It is clear from Figure 3 that the number of Fitness Function evaluations for PPSO and SPSO do not increase significantly in comparison to that of LyPSO and classical PSO algorithms.

Figure 3. Variation in number of fitness function evaluations with function dimension: (a) Sphere function. (b) Rosenbrock’s function. (c) Step function. (d) Schwefel’s Problem 1.2. (e) Rastrigin’s function. (f) Ackley’s function. (g) Griewank’s function. (h) Salomon’s function.

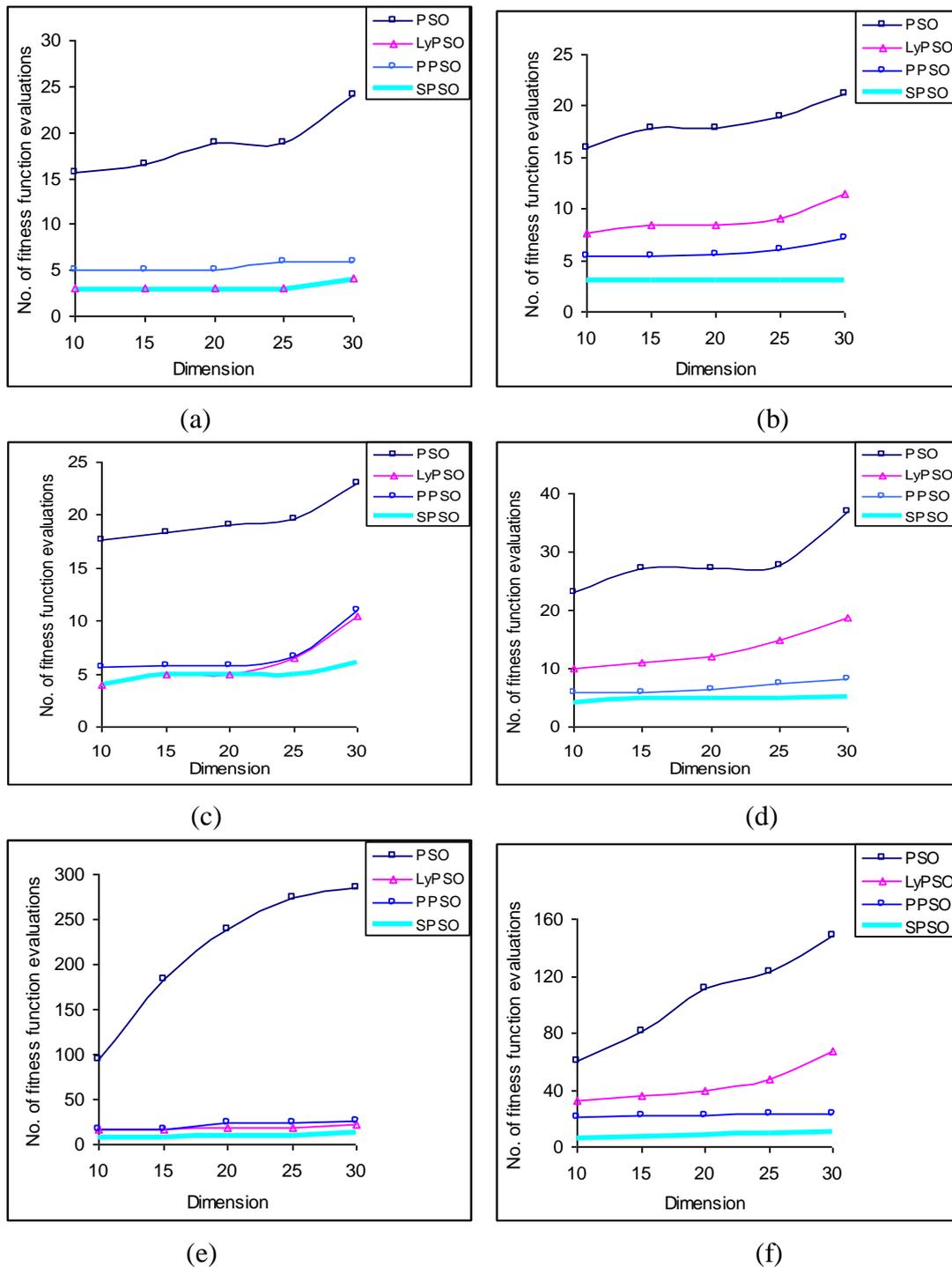


Figure 3. Cont.

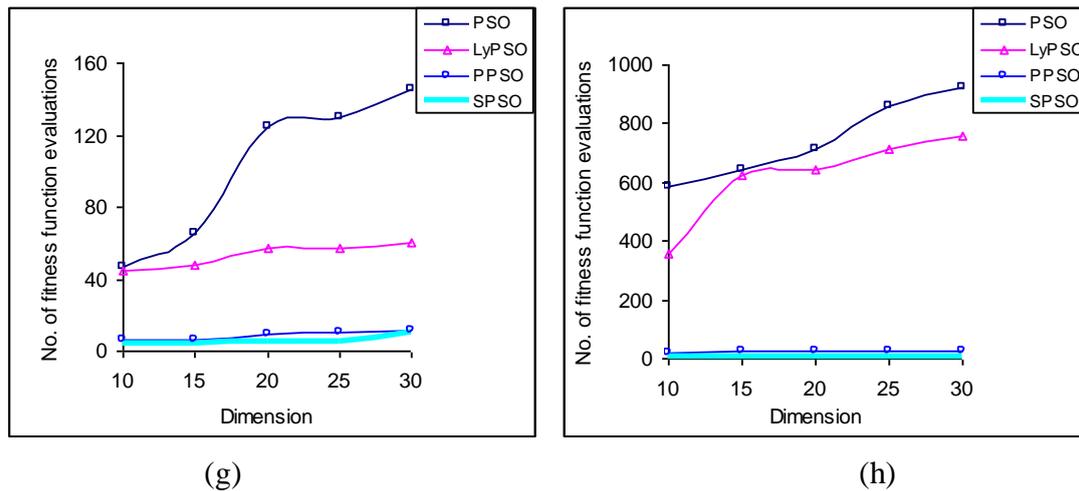


Table 8. Mean convergence time of the benchmarks over 30-dimensions.

Function Name	Dimension	Mean Convergence Time (in seconds)			
		Classical PSO	LyPSO	PPSO	SPSO
Sphere Function	30	24.0	4.8	6.0	4.8
Rosenbrock's Function	30	21.1	11.5	7.2	5.9
Step Function	30	23.0	10.5	11.0	6.6
Schwefel's Problem 1.2	30	36.8	18.6	8.3	4.3
Rastrigin's Function	30	563.5	23.0	26.0	9.0
Ackley's Function	30	148.9	67.2	23.2	10.9
Griewank's Function	30	172.1	60.5	11.5	8.0
Salomon's function	30	925.9	758.7	28.4	17.2

The PPSO, SPSO, LyPSO and PSO algorithms have been executed on eight benchmark functions, and for each algorithm the average of the convergence time for 50 independent runs to meet the error limit for individual function as specified in Table 6 is recorded in Table 8. It is clear from this Table that the mean convergence time of the SPSO is less than that of PPSO. The mean convergence time of PPSO is less than that of LyPSO, and the latter is less than the mean convergence time of classical PSO. The above phenomena is true for all benchmark functions except the sphere, where the LyPSO and SPSO gives identical results because of same functional form in the SPSO and LyPSO dynamics.

4. Conclusions

Classical g-best PSO has a proven impact in optimization of multi-modal nonlinear objective functions. However, for many nonlinear continuous multi-modal functions, where partial derivatives with respect to objective function variables exist, classical g-best PSO is not very efficient as it does not utilize gradient information of the search landscape. The paper bridges the gap between gradient-free and gradient-based optimization algorithm. It does not truly utilize gradient information of the search space, but it requires the background information that the gradient of the surface exists. When the

prerequisite knowledge about the search space is known, we extend the classical g-best PSO algorithm by the principles outlined in the paper.

Three alternative approaches to improve the speed of convergence of the PSO dynamics over continuous fitness landscapes is discussed in the paper. The first approach attempted to replace the inertial term in the dynamics by a factor that ensures asymptotic stability of the PSO dynamics. Construction of such dynamics presumes the characteristics of the surface being Lyapunov-like. This, however, is not a very restrictive assumption as many multi-modal surfaces support the conditions for Lyapunov function. On the contrary, the Lyapunov-based extension, even without local and global attractors, has a natural tendency to move towards optima on the surface. The convergence of the algorithm to local and global optima, however, is controlled by the presence of attractors in the PSO dynamics.

The second alternative approach to make the PSO smarter was derived from the Lyapunov-based formulation, just by noting that the Lyapunov-based dynamics includes a factor of negatively weighted position of the particle. Incorporation of this new term to the existing velocity adaptation rule classical PSO gives birth to the second alternative form of the extended PSO dynamics. The resulting dynamics has been found to have asymptotic stability for a selective range of $\omega < 1$, *i.e.*, same as in classical PSO. The third extension lies in replacement of the inertial term by the negative position of the particle itself. A random factor is attached to this term to maintain explorative power of the PSO dynamics to avoid its premature convergence. Computer simulations undertaken ensure that the third alternative form of extended PSO dynamics results in significant improvement in convergence time and accuracy compared to the results obtained by the first and second attempt. However, all three approaches outperform the classical PSO dynamics from the point of view of the convergence time and accuracy.

Future research efforts will focus on the extensions of Lyapunov-based dynamics of gbest PSO for handling non-continuous multi-modal functions. The particle dynamics will be combined with an estimate of the gradient of the function, instead of using any analytical expression of the partial derivatives of the objective function. Also the Lyapunov-based particle dynamics will be examined in context to non-linearly constrained optimization problems, where only a portion of the search space will be used to generate feasible optimal solutions.

Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the HNRC (Home Network Research Center)—ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0035) and also partially supported by National Research Foundation of Korea Grant funded by the Korean Government (2009-0076290).

References

1. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In *Proceedings of the 6th International Symposium on Micromachine Human Science*, Nagoya, Japan, October, 1995; pp. 39–43.

2. van den Bargh, F.; Engelbrecht, A.P. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comp.* **2004**, *8*, 225–239.
3. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In *Proceedings of IEEE Congress on Evolutionary Computing (CEC'98)*, Anchorage, AK, USA, May, 1998.
4. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73.
5. Eberhart, R.C.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computing (CEC2000)*, San Diego, CA, USA, July, 2000; pp. 84–89.
6. Angeline, P.J. Using selection to improve particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computing*, Anchorage, AK, USA, May, 1998; pp. 84–89.
7. Sukanathan, P.N. Particle swarm optimizer with neighborhood operator. In *Proceedings of IEEE Congress on Evolutionary Computing*, Piscataway, NJ, USA, July, 1999; pp. 1958–1962.
8. Eberhart, R.C.; Hu, X. Adaptive particle swarm optimization: detection and response to dynamic systems. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, May, 2002; pp. 1666–1670.
9. Parsopoulos, K.; Vrahatis, M. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 211–224.
10. Ratnaweera, A.; Halgamuge, S.; Watson, H. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255.
11. Løvberg, M.; Rasmussen, T.; Krink, T. Hybrid particle swarm optimizer with breeding and subpopulations. In *Proceedings of the 3rd Genetic and Evolutionary Computation Conference (GECCO'01)*, San Francisco, CA, USA, July, 2001; pp 469–476.
12. Miranda, V.; Fonseca, N. New evolutionary particle swarm algorithm (eps) applied to voltage/VAR control. In *Proceedings of the 14th Power Systems Computation Conference (PSCC'02)*, Seville, Spain, June, 2002.
13. Løvberg, M.; Krink, T. Extending particle swarms with self-organized criticality. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, May, 2002.
14. Blackwell, T.; Bentley, P.J. Don't push me! Collision-avoiding swarms. In *Proceedings of IEEE Congress on Evolutionary Computation*, Honolulu, HI, USA, May, 2002; pp. 1691–1696.
15. Krink, T.; Vesterstrøm, J. S.; Rigel, J. Particle swarm optimization with spatial particle explosion. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, May, 2002.
16. Xie, X.; Zhang, W.; Yang, Z. A dissipative particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, May, 2002.
17. Hendtlass, T.; Randall, M. A survey of ant colony and particle swarm metaheuristics and their application to discrete optimization problems. In *Proceedings of the Inaugural Workshop on Artificial Life (AL'01)*, Adelaide, Australia, December, 2001; pp. 15–25.
18. Coello, C.A.C.; Lechuga, M.S. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, May, 2002; pp.1051–1056.

19. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling Multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255.
20. Agrawal, S.; Panigrahi, B.K.; Tiwari, M.K. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. *IEEE Trans. Evol. Comput.* **2008**, *12*, 529–541.
21. Kuo, B.C. *Automatic Control Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1987.
22. Kalman, R.E.; Bertram, J.E. Control systems analysis and design via the second method of liapunov: II, discrete time systems. *Trans. ASME J. Basic Eng. D* **1960**, *3*, 371–400.
23. Hahn, W. *Theory and Application of Liapunov's Direct Method*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1963.
24. Ogata, K. *Modern Control Engineering*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1990.
25. Angeline, P. Evolutionary optimization verses particle swarm optimization: Philosophy and the performance difference. In *Proceedings of 7th International Conference on Evolutionary Programming*, San Diego, CA, USA, March, 1998; pp. 600–610.

© 2009 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).