

An Adaptive Particle Swarm Optimizer Using Balanced Explorative and Exploitative Behaviors

Sayan Ghosh¹, Debarati Kundu¹, Kaushik Suresh¹, Swagatam Das¹ and Ajith Abraham²

¹Department of Electronics and Telecommunication Engineering
Jadavpur University, Kolkata, India

²Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology,
Trondheim, Norway. Email: ajith.abraham@ieee.org

Abstract- Particle Swarm Optimization (PSO) has recently emerged as a nature inspired algorithm for real parameter optimization. This article describes a method for improving the final accuracy and the convergence speed of PSO by adding a new coefficient to the position updating equation and modulating the inertia weight. This work also mathematically analyzes the effect of this modification on the PSO algorithm. The new algorithm has been shown to be statistically significantly better than four recent variants of PSO on a six-functions test-suite.

1. Introduction

The concept of particle swarms, although initially introduced for simulating human social behavior, has become very popular these days as an efficient means of intelligent search and optimization. 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 swarming behavior of insects, and also draws inspiration from the boid's method of Craig Reynolds and socio-cognition [2]. Particles are conceptual entities, which search through a multi-dimensional search space. At any particular instant, each particle has a position and velocity. The position vector of a particle with respect to the origin of the search space represents a trial solution to the search problem.

PSO is however not free from false and/or premature convergence, especially over multimodal fitness landscapes. In this article, we describe a new variant of the basic PSO, which improves the performance of the algorithm. A momentum factor has been added to the position updating equation of the classical PSO, which gives greater mobility to the particles even when their velocities become very low due to false convergence to some local minima. Also, the inertia weight has been modulated.

The swarm dynamics of PSO have been mathematically analyzed in several papers since its inception. Ozcan and Mohan [3,4] analyzed the dynamics of the PSO by assuming constant values of inertia weight and acceleration coefficients, and by solving the relevant

recurrence relation in time for a single particle. The same assumptions were made by Clerc and Kennedy [5], who analysed the dynamics of a single particle on basis of the state-space representation of its velocity and position. Jiang *et al.* [6] applied stochastic techniques to derive the conditions of convergence for a particle swarm, where the inertia weight and acceleration coefficients were not assumed constant, but were treated as random variables with uniform distribution.

In this paper, we analyze the dynamics of a particle and show how the addition of a mobility factor promotes fast convergence and prevents premature convergence. We apply the mathematics of linear recurrence relations to analyze the time-behavior of a particle and calculate the particular solution. From the formulation of the particular solution, we show how the particles are prevented from converging at false optima.

2. Inertia-adaptive PSO Algorithm

A. The Classical PSO Algorithm

The classical PSO starts with the random initialization of a population of candidate solutions (particles) over the fitness landscape. However, unlike other evolutionary computing techniques, PSO uses no direct recombination of genetic material between individuals during the search. Rather it works depending on the social behavior of the particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual towards its own best position and toward the best particle of the entire swarm at each time-step (generation).

In a D -dimensional search space, the position vector of the i -th particle is given by $\vec{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ and velocity of the i -th particle is given by $\vec{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$. Positions and velocities are adjusted and the objective function to be optimized $f(\vec{X}_i)$ is evaluated with the new coordinates at each time-step. The velocity and position update equations for the d -th dimension of the i -th particle in the swarm may be represented as:

$$v_{i,d,t} = v_{i,d,t-1} + C_1 * rand_1 * (pbest_{i,d} - x_{i,d,t-1}) + C_2 * rand_2 * (gbest_d - x_{i,d,t-1})$$

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

where $rand_1$ and $rand_2$ are random positive numbers uniformly distributed in (0,1) and are drawn anew for each dimension of each particle. $pbest$ is the personal best solution found so far by an individual particle while $gbest$ represents the fittest particle found so far by the entire community. The first term in the velocity updating formula is referred to as the ‘cognitive part’. The last term of the same formula is interpreted as the ‘social part’, which represents how an individual particle is influenced by the other members of its society. C_1 and C_2 are called acceleration coefficients and they determine the relative influences of the cognitive and social parts on the velocity of the particle. The particle’s velocity is clamped to a maximum value $\vec{V}_{max} = [v_{max,1}, v_{max,2}, \dots, v_{max,D}]^T$. If in d -th dimension, $|v_{i,d}|$ exceeds $v_{max,d}$ specified by the user, then the velocity of that dimension is assigned to $sign(v_{i,d}) * v_{max,d}$, where $sign(x)$ is the triple-valued signum function.

B. Modifications to the Classical PSO Algorithm

Premature convergence occurs when the positions of the most of the particles of the swarm stop changing over successive iterations although the global optimum remains undiscovered. This may happen if the swarm uses a small inertia weight [7] or a constriction coefficient [5]. From the basic equations of PSO, we see that if $v_{i,d}$ is small and in addition to that $|pbest_{i,d} - x_{i,d}|$ and $|gbest_d - x_{i,d}|$ are small enough, $v_{i,d}$ cannot attain a large value in the upcoming generations. That would mean a loss of exploration power and may occur even at an early stage of the search process, when the particle itself is the global best causing $|pbest_{i,d} - x_{i,d}|$ and $|gbest_d - x_{i,d}|$ to be zero. $v_{i,d}$ gets damped quickly with the ratio ω . Also the swarm suffers from loss of diversity in later generations if $pbest$ and $gbest$ are close enough [8]. The first of these modifications involves modulation of the inertia factor ω according to distance of the particles of a particular generation from the global best. The value of ω for each particle is given by:

$$\omega = \omega_0 \cdot \left(1 - \frac{dist_i}{max_dist} \right), \quad (2)$$

where $\omega_0 = rand(0.5,1)$, $dist_i$ is the current Euclidean distance of i -th particle from the global best i. e.

$$dist_i = \left(\sum_{d=1}^D (gbest_d - x_{i,d})^2 \right)^{1/2} \quad (3)$$

and max_dist is the maximum distance of a particle from the global best in that generation i.e.

$$max_dist = \arg \max_i (dist_i) \quad (4)$$

This modulation of the inertia factor ensures that in case of particles that have moved away from the global best, the effect of attraction towards global best will predominate. Premature convergence is a bane of the PSO and in later stages the particles tend to get stuck in local minima, especially in case of multi-modal fitness landscapes and several real life problems. To ensure that the particle has mobility in the later stages, the position update equation is modified as follows:

$$x_{i,d,t} = (1 - \rho)x_{i,d,t-1} + v_{i,d,t} \quad (5)$$

where ρ is a uniformly distributed random number in the range (-0.25, 0.25). From now on, we shall refer to this new algorithm as IAPSO (Inertia-adaptive PSO).

3. Mathematical Model of IAPSO

A. Analysis of the Dynamics of a Particle

In this Section, we show mathematically how the inclusion of a momentum factor ρ in the position update equation helps us to tune the rate of convergence of the PSO. Thus by varying ρ , we can achieve a desired level of convergence. From the same mathematical analysis we go on to prove that premature convergence is discouraged in the new algorithm, thus giving the particles mobility even at later stages of convergence. For simplicity, we assume that only one particle is moving in a one-dimensional space. The analysis is done in discrete time domain. The velocity of the particle at a time instant ‘ t ’ is denoted by v_t and the cartesian coordinates of the position is denoted by x_t . In a discrete time domain, the position update equation involving v_t and x_t is:

$$v(t+1) = \omega v_t + \phi_1 (pbest - x_t) + \phi_2 (gbest - x_t) \quad (6)$$

$$= \omega v_t + (\phi_1 pbest + \phi_2 gbest) - (\phi_1 + \phi_2)x_t$$

where ω is the inertia factor and ϕ_1 and ϕ_2 are the acceleration coefficients which can be lumped into a single coefficient $\phi = \phi_1 + \phi_2$. We assume that ω , ϕ_1 and ϕ_2 are constant throughout the entire time for which the analysis is made. $pbest$ denotes locally best position of the particle and $gbest$ denotes its globally best position found so far. We further assume that $pbest$ and $gbest$ do not change with time. This enables us to determine how a particle converges towards static optimal positions. We can rewrite equation (6) as follows:

$$v_{t+1} = \omega v_t + \phi [p - x_t] \quad (7)$$

$$\text{where } p = \frac{\phi_1 pbest + \phi_2 gbest}{\phi_1 + \phi_2}$$

The position update equation of a particle is:

$$x_{t+1} = \rho'x_t + v_{t+1}$$

where $\rho' = 1 - \rho$; ρ is the momentum factor. This factor is also assumed to be constant throughout.

We now introduce another state variable y_t where $y_t = p - x_t$. Since the local and globally best positions do not change with time, p is also constant with time. We now substitute y_t in place of $(p - x_t)$ in the velocity update equation to obtain:

$$v_{t+1} = \omega v_t + \phi y_t \quad (8)$$

We now carry out the same substitution in the position update equation.

$$\begin{aligned} x_{t+1} &= (\rho' - \phi)x_t + \omega v_t + \phi p \\ \Rightarrow p - y_{t+1} &= (\rho' - \phi)(p - y_t) + \omega v_t + \phi p \\ \Rightarrow p - y_{t+1} &= \rho'(p - y_t) + \phi y_t + \omega v_t \\ \Rightarrow y_{t+1} &= (\rho' - \phi)y_t - \omega v_t + p(1 - \rho') \end{aligned} \quad (9)$$

We introduce the shift operator E in our analysis, so that $E v_t = v_{t+1}$. Similarly $E x_t = x_{t+1}$. The velocity and position update equations, rewritten using the shift operator are:

$$\begin{aligned} E y_t &= (\rho' - \phi)y_t - \omega v_t + p(1 - \rho') \\ E v_t &= \omega v_t + \phi y_t \end{aligned} \quad (10), (11)$$

Our next task is to combine equations (10) and (11) into a single recurrence relation in v_t .

$$\text{Now, } E v_t = \omega v_t + \phi y_t \Rightarrow E^2 v_t = \omega E v_t + \phi E y_t$$

$$\text{Also, } E y_t = (\rho' - \phi)y_t - \omega v_t + p(1 - \rho')$$

$$\begin{aligned} \therefore E^2 v_t &= \omega E v_t + \phi(\rho' - \phi)y_t - \omega \phi v_t + p\phi(1 - \rho') \\ &= \omega E v_t + (\rho' - \phi)(E v_t - \omega v_t) - \omega \phi v_t + p\phi(1 - \rho') \\ &= \omega E v_t + (\rho' - \phi)E v_t - \omega \rho' v_t + \omega \phi v_t - \omega \phi v_t + p\phi(1 - \rho') \\ &= \omega E v_t + (\rho' - \phi)E v_t - \omega \rho' v_t + p\phi(1 - \rho') \\ \Rightarrow E^2 v_t + (\phi - \rho' - \omega)E v_t + \rho' \omega v_t &= p\phi(1 - \rho') \end{aligned}$$

This is the linear recurrence relation to be solved. It has constant coefficients. Similar to the procedure for solving a differential equation we find the:

1. complementary function $c(t)$ by solving:

$$(E^2 + (\phi - \rho' - \omega)E + \rho' \omega)v_t = 0 \quad (12)$$

2. particular solution (P.S) by finding:

$$\frac{1}{E^2 + (\phi - \rho' - \omega)E + \rho' \omega} p\phi(1 - \rho')$$

B. Solution of the Recurrence Relation

It is the complementary function, which describes the behavior of the system (unstable, stable, oscillatory). If the complementary function converges to an infinitesimally small value with time, then the particular solution gives the ultimate location of the particle.

The solution of equation (12) is:

$$E = \frac{(\rho' + \omega - \phi) \pm \sqrt{(\rho' + \omega - \phi)^2 - 4\rho'\omega}}{2}$$

We define the roots of the equation as e_1 and e_2 .

$$e_1 = \frac{(\rho' + \omega - \phi) + \sqrt{(\rho' + \omega - \phi)^2 - 4\rho'\omega}}{2}$$

$$e_2 = \frac{(\rho' + \omega - \phi) - \sqrt{(\rho' + \omega - \phi)^2 - 4\rho'\omega}}{2}$$

Depending on the nature of the roots, three classes of solutions are possible.

Class 1: The discriminant D is zero, i.e.

$$(\rho' + \omega - \phi)^2 = 4\rho'\omega \quad (13)$$

Then, $e_1 = e_2 = \frac{\rho' + \omega - \phi}{2} = \sqrt{\rho'\omega}$, and the

complementary function is :

$$c(t) = (c_1 + c_2 t)e_1^t \text{ where } c_1 \text{ and } c_2 \text{ are arbitrary constants.}$$

Thus to obtain critically damped motion of the particle, only two values of ρ' are permissible, obtained by solving equation (13). By tuning ρ' appropriately, we can set the motion to critical damping. Also we have to ensure $0 < \rho'\omega < 1$ for the solved values of ρ' .

Class 2: The discriminant D is positive i.e.

$$(\rho' + \omega - \phi)^2 > 4\rho'\omega$$

Then e_1 and e_2 are real and distinct.

Then $c(t) = c_1 e_1^t + c_2 e_2^t$ where c_1 and c_2 are arbitrary constants. The C.F converges iff $|e_1| < 1$ and $|e_2| < 1$.

Class 3: The discriminant D is negative, i.e.

$$(\rho' + \omega - \phi)^2 < 4\rho'\omega$$

Then e_1 and e_2 are complex conjugate solutions. Then

Suppose $e_1 = \alpha + i\beta$ and $e_2 = \alpha - i\beta$ where

$$\alpha = \frac{\rho' + \omega - \phi}{2} \text{ and } \beta = \frac{\sqrt{4\rho'\omega - (\rho' + \omega - \phi)^2}}{2}$$

The complementary function $c(t)$ is given by

$r^t [c_1 \cos(t\theta) + c_2 \sin(t\theta)]$ where the values of

r and θ are given by $r = \sqrt{\alpha^2 + \beta^2}$ and $\theta = \tan^{-1} \frac{\beta}{\alpha}$.

c_1 and c_2 are arbitrary constants. Thus the particle 'surfs' on

a foundation of sine waves in this case. The period of oscillation is given by:

$$\theta_p = \frac{2k\pi}{t}; k \in \{1, 2, \dots, t-1\}. \text{ For convergence we require}$$

$$|r| < 1 \Rightarrow \alpha^2 + \beta^2 < 1 \Rightarrow \rho' \omega < 1$$

Thus in this case if the parameter ρ' is introduced, r reduces further, thus increasing the speed of convergence.

Thus, in both classes 1 and 2 of solutions we note that the convergence rate is entirely dependent on the parameter ρ' . The particular solution is computed as:

$$\begin{aligned} & \frac{1}{E^2 + (\phi - \rho' - \omega)E + \rho'\omega} p\phi(1 - \rho') \\ &= \frac{1}{(\Delta + 1)^2 + (\phi - \rho' - \omega)(\Delta + 1) + \rho'\omega} p\phi(1 - \rho') \\ &= \frac{p\phi(1 - \rho')}{\phi + (1 - \rho')(1 - \omega)} \end{aligned}$$

Hence, the complete solution of the velocity is given by:

$$v_t = c(t) + \frac{p\phi(1 - \rho')}{\phi + (1 - \rho')(1 - \omega)}, \quad (14)$$

where $c(t)$ is given by the three different classes of solution described earlier. The complementary function $c(t)$ converges towards zero for $t \rightarrow \infty$, subject to certain conditions in ρ' , ω and ϕ . Then

$$v_t \rightarrow \frac{p\phi(1 - \rho')}{\phi + (1 - \rho')(1 - \omega)} \text{ as } t \rightarrow \infty. \text{ Hence the particular}$$

solution defines only the target, and not the manner in which v_t reaches the target value

C. Prevention of Premature Convergence

We begin our analysis from the time instant when v_t has converged and reached its constant value. The constant value of v_t is now denoted by v_0

$$\text{where } v_0 = \frac{p\phi(1 - \rho')}{\phi + (1 - \rho')(1 - \omega)}. \text{ Now from equation (14) we}$$

have

$$v_{t+1} = c(t+1) + \frac{p\phi(1 - \rho')}{\phi + (1 - \rho')(1 - \omega)} \quad (15)$$

Substituting the derived relations for v_t and v_{t+1} in equation (14) we get :

$$c(t+1) + v_0 = \omega c(t) + \omega v_0 + \phi y_t$$

$$\Rightarrow \lim_{t \rightarrow \infty} c(t+1) + v_0 = \omega \lim_{t \rightarrow \infty} c(t) + \omega v_0 + \phi \lim_{t \rightarrow \infty} y_t$$

$$\Rightarrow \lim_{t \rightarrow \infty} y_t = \frac{1}{\phi} v_0 (1 - \omega) \quad \because \lim_{t \rightarrow \infty} c(t+1) = \lim_{t \rightarrow \infty} c(t) = 0$$

$$\Rightarrow \lim_{t \rightarrow \infty} y_t = \frac{p(1 - \rho')(1 - \omega)}{\phi + (1 - \rho')(1 - \omega)}$$

Thus the final value of y_t is a finite real number. For the classical PSO, $\rho' = 1$ always, so y_t becomes zero with time. This corresponds to the case of premature convergence. In the actual IAPSO, ρ' is not kept constant but is initialized to a random value at every generation, and so y_t does not remain constant. In this way, premature convergence is avoided.

D. Tuning of the Parameter ρ'

In the previous section we have shown that we can select appropriate values of the parameter ρ' to control the speed of convergence. Our selection depends on the values of ω and ϕ as well. We have seen that to ensure convergence, the discriminant D of the auxiliary equation (12) should be less than or equal to zero. Moreover the condition $|\rho'\omega| < 1$ should be ensured to maintain convergence. We first solve the equation $D=0$ and the inequations $D>0$ and $D<0$. Rewriting the equation in terms of ρ' , ω and ϕ we get:

$$(\rho' + \omega - \phi)^2 - 4\rho'\omega = 0$$

$$\Rightarrow \rho'^2 + \omega^2 + \phi^2 - 2\rho'\omega - 2\omega\phi - 2\rho'\phi = 0$$

$$\Rightarrow \rho'^2 - 2\rho'(\omega + \phi) + (\omega - \phi)^2 = 0$$

The roots of the above equation are $\rho'_1 = (\sqrt{\omega} - \sqrt{\phi})^2$ and $\rho'_2 = (\sqrt{\omega} + \sqrt{\phi})^2$. The solution of $D>0$ is given by $\rho' < \rho'_1$ or $\rho' > \rho'_2$. Similarly the solution of $D<0$ is given by $\rho'_1 < \rho' < \rho'_2$.

We now calculate values of ρ' for constant values of $\omega = \omega_0$ and $\phi = \phi_0$. For class 1 of solutions, we require $\rho' = (\sqrt{\omega_0} - \sqrt{\phi_0})^2$ or $\rho' = (\sqrt{\omega_0} + \sqrt{\phi_0})^2$, subject to the condition $|\rho'\omega_0| < 1$. To analyze class 2 of solutions, we require $\rho' > (\sqrt{\omega_0} + \sqrt{\phi_0})^2$ or $\rho' < (\sqrt{\omega_0} - \sqrt{\phi_0})^2$ and $\max(|e_1|, |e_2|) < 1$. By analyzing e_1 and e_2 as functions of ρ' we can show that the latter condition is satisfied when:

$$\max\left(|\omega_0 + \sqrt{\omega_0\phi_0}|, |\omega_0 - \sqrt{\omega_0\phi_0}|\right) < 1 \quad (16)$$

For class 3 of solutions we require $(\sqrt{\omega_0} - \sqrt{\phi_0})^2 < \rho' < (\sqrt{\omega_0} + \sqrt{\phi_0})^2$ and $|\rho'\omega_0| < 1$.

We now apply our analysis to the case $\omega_0 = 0.765$ and $\phi_0 = 4.0$ (the values have been taken from the standard literature on PSO).

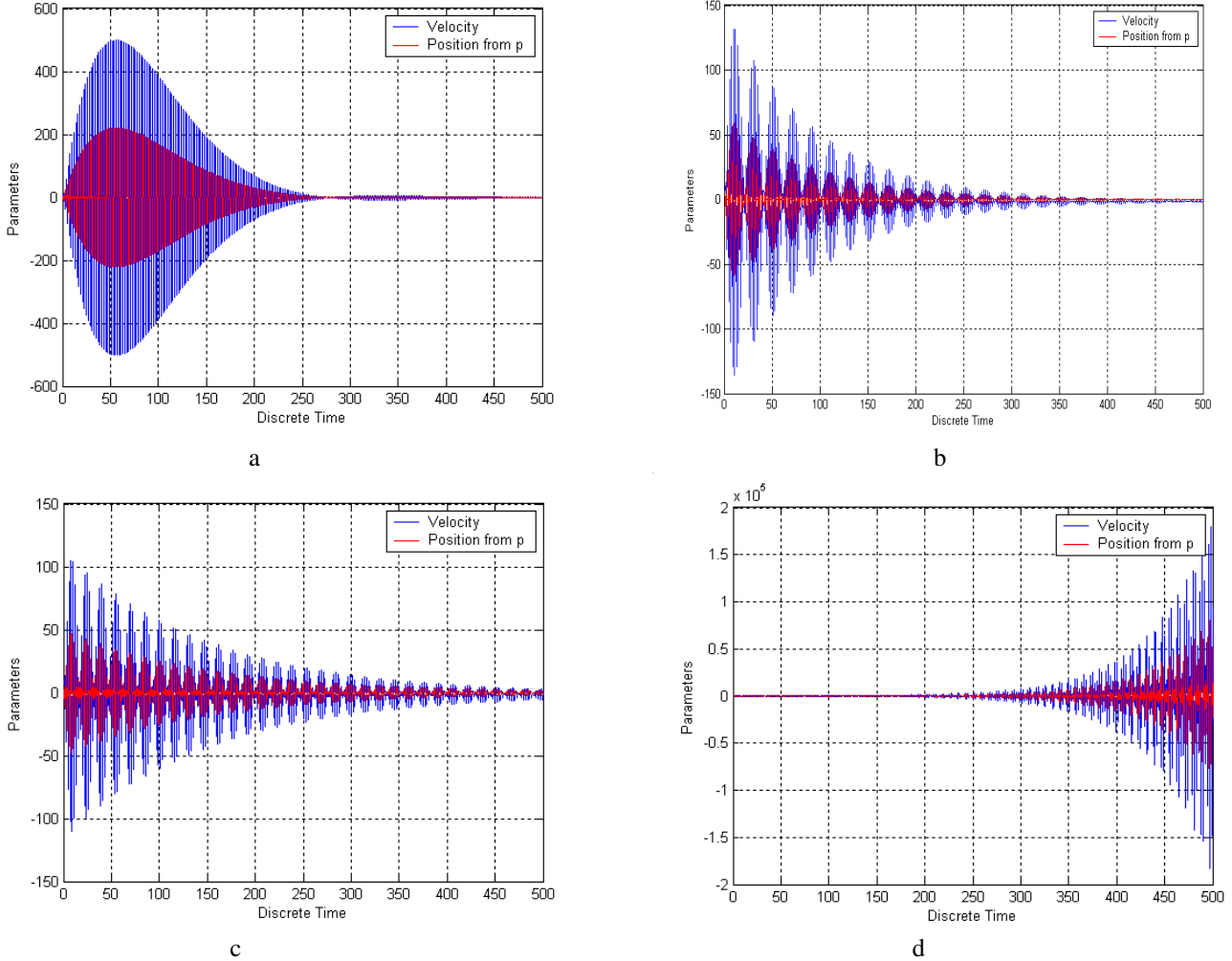


Figure 1. The time-behavior of velocity and position of the particle for various values of ρ' , when $\omega_0 = 0.765$ and $\phi_0 = 4.0$: (a) $\rho' = -0.2665$ (critical convergence) (b) $\rho' = -0.275$ (fast convergence) (c) $\rho' = -0.29$ (slow convergence) and (d) $\rho' = -0.35$ (non-convergence)

For class 1 of solutions, the value of ρ' is given by $\rho' = -0.2665$ (the other solution does not ensure convergence). For class 2 of solutions the range of ρ' is $\rho' > -0.2665$ or $\rho' < -0.30718$. No convergence is possible for any value of ρ' in this case, since condition (16) is not satisfied. For class 3 of solutions we require $-0.30718 < \rho' < -0.2665$. The time-behavior of the position and velocity of the particle for various values of ρ' are shown in Fig 1. From the figure, it can be seen that various rates of convergence can be obtained by appropriately adjusting the value of ρ' .

4. Experimental Setup

A. Benchmark functions

We have used eight well-known benchmarks [9] to evaluate the performance of the proposed algorithm. The benchmark functions are: f_1 : Rosenbrock's function; f_2 : Rastrigin's function; f_3 : Ackley's function; f_4 : Weierstrass function; f_5 : Griewank's function; f_6 : Generalized Penalized function 1 (GP-1). Here the proposed algorithm has been compared with the classical PSO algorithm and four of its significant variants over these benchmark functions. In Table 1, D represents the number of dimensions (we used $D = 30$ and 60). An asymmetrical initialization procedure has been used here following the work reported in [10].

Table 1. Average and the standard deviation of the best-of-run solution for the 50 independent runs and the success rate tested on f_1 to f_6

Function	D	FEs	Mean Best Value (Standard Deviation)						Statistical Significance Level
			BPSO	PSO-TVIW	MPSO- TVAC	HPSO- TVAC	CLPSO	IAPSO	
f_1	30	5×10^5	2.063e+004 (6.78e+004)	2.196e+002 (8.45e+001)	6.8372e+001 (4.75e+00)	7.332e+001 (7.13e+001)	5.670e+001 (5.16e+001)	2.8676e+001 (2.68e-001)	-
	60	1×10^6	4.1336e+003 (3.69e+003)	7.0931e+002 (6.22e+001)	3.8274e+002 (2.378e+001)	1.9451e+002 (3.94e+002)	1.177e+002 (8.69e+001)	4.3567e+001 (1.06e-001)	-
f_2	30	5×10^5	5.6352e+001 (3.54e+001)	4.2455e+001 (1.96e+001)	9.5278e+001 (9.72e+00)	3.9426e+001 (3.10e+001)	1.3107e-001 (3.24e-001)	1.5713e-053 (2.07e-060)	+
	60	1×10^6	1.2245e+002 (5.18e+001)	1.1283e+000 (4.46e-01)	3.7649e+001 (4.27e+00)	6.8186e+001 (4.13e+001)	8.4291e-001 (1.53e+000)	0.00e+000 (0.00e+000)	+
f_3	30	5×10^5	4.73e+000 (3.03e+000)	4.0364e-001 (2.81e-003)	7.94504e-002 (8.03e-003)	3.6982e+000 (1.95e-001)	2.7445e-003 (1.73e-003)	5.8924e-016 (0.00e+000)	+
	60	1×10^6	8.5297e+000 (5.23e+000)	1.0222e+000 (1.82e-001)	5.2724e-001 (4.63e-007)	5.56e+000 (3.08e+000)	2.4501e-002 (1.33e-002)	2.9655e-015 (5.68e+020)	+
f_4	30	5×10^5	2.04e+001 (1.43e+001)	3.9716e-001 (6.39e-002)	2.8962e-001 (2.25e-002)	7.6843e+000 (7.30e+000)	3.9812e-008 (1.95e-009)	3.0300-013 (1.60e-020)	+
	60	1×10^6	1.79e+001 (7.31e+000)	3.0835e+001 (4.73e-001)	5.2184e-001 (2.94e-004)	1.3732e+001 (5.63e+000)	1.1403e-006 (3.26e-003)	3.9304e-010 (1.54e-009)	-
f_5	30	5×10^5	9.5294e-001 (2.42e-001)	2.0621e-002 (5.58e-03)	9.8035e-001 (6.80e-03)	1.8235e-002 (2.93e-002)	1.1435e-003 (1.74e-003)	0.00e+000 (0.00e+000)	+
	60	1×10^6	4.7364e+000 (1.77e+001)	4.0832e-001 (5.42e-002)	6.76249e-001 (4.27e-001)	1.2065e-002 (2.14e-003)	6.9734e-003 (4.05e-003)	0.00e+000 (0.00e+000)	+
f_6	30	5×10^5	5.9242e+000 (4.88e+000)	1.0045e+001 (4.32e-001)	4.8605e+001 (1.08e+000)	4.5170e+000 (3.82e+000)	9.0408e-001 (1.77e-005)	1.1740e-001 (9.73e-004)	+
	60	1×10^6	2.64e+001 (1.57e+001)	1.0400e+001 (8.54e-001)	5.81493e-001 (1.08e-002)	1.3531e+001 (7.77e+000)	1.8425e-001 (1.31e-001)	5.1963e-001 (2.61e-001)	-

B. Algorithms Compared

Simulations were carried out to obtain a comparative performance analysis of the proposed IAPSO algorithm with respect to: (a) the basic PSO (BPSO) with constant inertia weight (b) PSO-TVIW [11] (c) HPSO-TVAC [12] (d) MPSO-TVAC [12], and (e) CLPSO [13]. In all the algorithms, for a particular trial, the same initial positions and velocities were set for all particles, so as to minimize the effect of randomness during comparison.

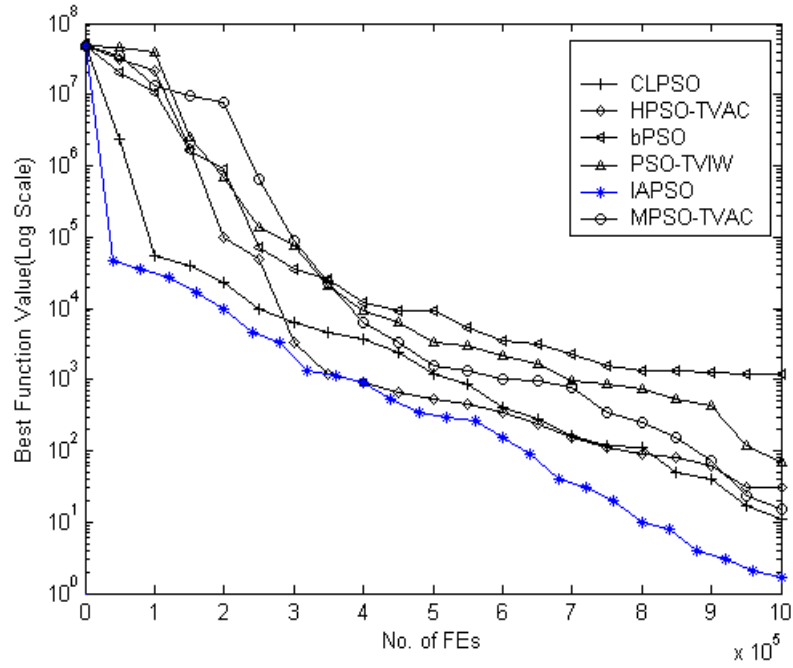
C. Simulation Strategy

To judge the accuracy of different PSO-variants, we first let each of them run until the number of function evaluations (FEs) exceed a given upper limit (which was fixed at 10^6) and record the final best fitness achieved by each algorithm. We employed the best set of parameters for all the considered algorithms, as found in the relevant literatures. We have set the population size equal to 40 particles. For IAPSO, we set $C_1 = C_2 = 2.00$ and

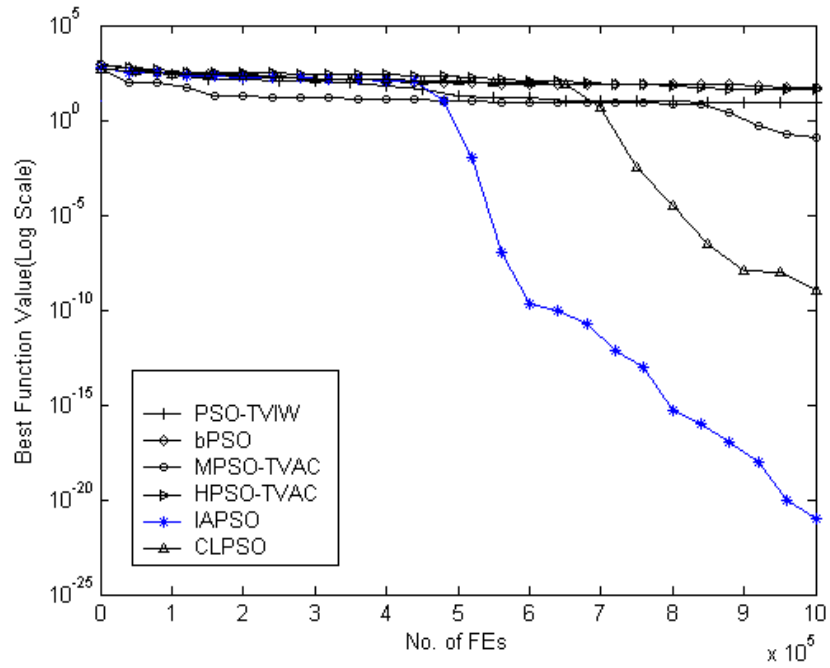
initialized ρ to a random value in the range $(-0.25, 0.25)$. For all particles, ρ is changed with every iteration. The value of ω is set according to equation (1) where ω_0 is assigned a random value in the range $(0.5, 1)$. All the algorithms discussed here have been developed from scratch in Visual C++ on a Pentium IV, 2.3 GHz PC, with 1024 KB cache and 2 GB of main memory in Windows XP environment.

5. Results and Discussions

The mean and the standard deviation (within parentheses) of the best-of-run solution for 50 independent runs of each of the six algorithms are presented in Table 1. Since all the algorithms start with the same initial population over each problem instance, we used paired t -tests to compare the means of the results produced by best and the second best algorithms. The 10-th column of Table 1 reports the statistical significance level of the difference of the means of best two algorithms.



a. Rosenbrock's function (f_1)



b. Weierstrass function (f_4)

Figure 2. Performance for functions f_1 and f_4

Note that here '+' indicates the t value of 49 degrees of freedom is significant at a 0.05 level of significance by two-tailed test, '-' means the difference of means is not statistically significant.

In Figure 2, we have graphically presented the rate of convergence of all algorithms over two test functions.

6. Conclusions

This work has presented a new, efficient PSO algorithm, which self-adapts the inertia weight over different fitness landscapes. The algorithm has been subjected to a mathematical analysis to show how the inclusion of a mobility factor improves convergence speed and prevents premature convergence. The new method has been compared against the basic PSO and four well-known PSO-variants, using a six-function test suite. It has been shown to outperform its nearest competitor in a statistically meaningful way for majority of the test cases. Since all the algorithms start with the same initial population, difference in their performances must be due to the difference in their internal operators and parameter values.

References

- [1] Kennedy, J., Eberhart, R. C.: (1995) Particle swarm optimization, In Proceedings of IEEE International conference on Neural Networks. 1942-1948.
- [2] Kennedy, J., Eberhart, R. C., and Shi, Y.: (2001) *Swarm Intelligence*. Morgan Kaufman, San Francisco, USA.
- [3] E. Ozcan, C.K. Mohan, Analysis of a simple particle swarm optimization system, in: *Intelligent Engineering Systems through Artificial Neural Networks*, 1998, pp. 253-258.
- [4] E. Ozcan, C.K. Mohan, Particle swarm optimization: surfing the waves, in: *Proc. IEEE Congress on Evolutionary Computation (CEC 1999)*, Washington, DC, USA, 1999, pp. 1939-1944.
- [5] Clerc, M. and Kennedy, J.: (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space, In *IEEE Transactions on Evolutionary Computation* 6(1): 58-73.
- [6] Jiang, M., Luo, Y.P., Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, *Information Processing Letters*, 102, 2007, 8-16
- [7] R. C. Eberhart, Y. Shi.: Particle swarm optimization: Developments, applications and resources, In *Proceedings of IEEE International Conference on Evolutionary Computation*, vol. 1 (2001), 81-86.
- [8] Xie, X., F. Zhang, W., J., and Yang, Z, L. Adaptive particle swarm optimization on individual level, In *Proceedings of International Conference on Signal Processing* (2002), 1215-1218.
- [9] Das, S., Konar, A., and Chakraborty, U. K.: (2005) Improving Particle Swarm Optimization with Differentially Perturbed Velocity, in *ACM-SIGEVO Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2005)*, Washington DC.
- [10] Angeline, P. J.: (1998) Evolutionary optimization versus particle swarm optimization: Philosophy and the performance difference, *Lecture Notes in Computer Science*, vol. 1447, Evolutionary Programming VII, 84-89.
- [11] Shi, Y. and Eberhart, R. C.: (1999) Empirical Study of particle swarm optimization, In *Proceedings of IEEE International Conference Evolutionary Computation*, Vol. 3 , 101-106.
- [12] Ratnaweera, A., Halgamuge, K. S., and Watson, H. C.: (2004) Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, In *IEEE Transactions on Evolutionary Computation* 8(3): 240-254.
- [13] Liang, J. J., Qin, A. K., Suganthan, P. N., and Baskar, S.: (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp. 281-295, 2006.