MultiCraft

International Journal of Engineering, Science and Technology
Vol. 8, No. 3, 2016, pp. 64-75

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY

www.ijest-ng.com
www.ajol.info/index.php/ijest

Cognitive and social information based PSO

D. P. Tripathi[1*], U. R. Jena[2]

[1*] *Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, INDIA*
[2] *Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, INDIA*
[*] *Corresponding Author: e-mail: dpt.tara@gmail.com, Tel: +91-9937397337*

Abstract

In 1995 swarm intelligence based PSO (Particle Swarm Optimization) has been designed and implemented for solving optimization problems. Since then many researchers have developed many versions, based upon its theoretical concept, technical aspects and parameters involve in the algorithm. In broad sense, every swarm updates its position based upon the knowledge of its initial velocity and accelerating components such as cognitive and social information. In this paper we have presented a reformed and modified concept of PSO with the thought that every swarm updates its position based upon cognitive and social environment knowledge only and the key aspect used here is that these parameters are no longer assumed to be accelerating components rather position components. This algorithm is termed by us as Cognitive and Social Information based PSO (CSI-PSO). The performance of CSI-PSO is validated by 23 benchmark functions and the empirical results clearly support the effectiveness of our concept.

Keywords: Particle Swarm, PSO, Swarm Theory, Benchmark functions

DOI: http://dx.doi.org/10.4314/ijest.v8i3.6

1. Introduction

Based upon the simulation of swarm of birds searching for food, Eberhart and Kennedy (1995) & Kennedy and Eberhart (1995) have proposed particle swarm optimization algorithm. In this algorithm, each bird act as a particle and update its position based upon its initial velocity, its personal last position visited i.e. cognitive component and information gather about the neighboring best position from social interaction i.e. social component. PSO has certain key merits compared to other evolutionary algorithms such as faster convergence rate in certain optimization problems (Kennedy and Eberhart, 2001), secondly less number of parameters to be vary are less. The major drawback of PSO is its sensitivity to fall into local optimum points while optimizing complex multimodal functions. However looking at its greater and easy implementation qualities researchers have shown a great interest in this algorithm. In last two decades different modification have been done in PSO algorithm and successfully applied in different applications such as neural network (Wei *et al*., 1998) and many more.

Inertia factor (w) has been inducted in the standard PSO to have a better control over the velocity of swarm particles and overall to get a better performance (Shi and Eberhart**,** 1998). Xu *et al.* (2012) has proposed a new incremental particle size PSO (IPPSO), where the size of the population increases with every iteration based upon the logistic human population growth. A fast convergence particle swarm optimization algorithm has been proposed in which a new parameter named particle mean dimension has been used to balance the diversity of location of each particles in the swarm to bring overall improvement in the performance of PSO (Sahu *et al.*, 2012). A new variant of PSO based on constriction factor approach where the algorithm ensures the convergence of the search procedure along with better solutions than standard PSO (Clerc and Kennedy, 2002).

Dynamic chaos has been incorporated with hybrid PSO to improve the efficiency of the algorithm (Liu *et al.*, 2005). For complex multidimensional problems, the PSO particles ability for stable convergence have been explored by Clerc and Kennedy (2002). Eberhart and Shi (1998) have compared the exploration and exploitation capacity of GA and PSO. Shi *et al.* (2005) have proposed a novel algorithm based on hybridization of PSO and genetic algorithm. Angeline (1998) has incorporated the technique of tournament selection with PSO where the properties of the current best particles were transferred to inferior ones. Suganthan (1999) proposed a variant of the algorithm known as local best (LBest) model, where the entire swarm population is divided into small neighborhood and each neighborhood maintains its own local best. This technique very well avoids trapping of the swarm in local optima solution but its convergence rate is very slow. Nickabadi *et al.* (2011) have proposed adaptive inertia weight PSO algorithm. Chatterjee and Siarry (2006) have introduced a new variant of PSO to reduce exploitation time to reach global optima and for improving the search capacity in multidimensional space nonlinear variation of inertia weight has been used. Alireza (2011) has improved the performance of PSO by mutating the particles of PSO adaptively and adding a dynamic behavioral inertia factor. Modares *et al.* (2010) have proposed an improved PSO technique for identification of parameters in chaotic dynamic systems. In multimodal optimization problems have many global optimum solutions can be found using NichePSO (Brits *et al*, 2007). Like PSO another evolutionary algorithm BSA (Civicioglu, 2013), which chooses the direction of an individual to move from any individuals of the last generation randomly while looking for global optima.

The PSO algorithm has successfully explored and exploited the number of different unimodal optimization problems (Shi and Eberhart, 1999). Van den Bergh and Engelbrecht (2006) in their work studied the trajectories of general swarm particles to include the effect of inertia term in PSO. Tripathi *et al.* (2007) in their novel approach changes the inertia weight parameter along with the coefficient of acceleration in each iteration to improvise the exploration capacity of PSO in search problem domain. Here we present a novel reformed modified version of PSO for optimization problems. We termed our propose algorithm as Cognitive and Social information based PSO (CSI-PSO) This paper is different from rest of the works in two aspects firstly, here we have completely eliminate the velocity concept of a swarm particle and assumes that each particle from its own position in search space using its personal and social information as position updating tool to updates its position. Secondly the capability of computation of finding global optimum of 23 nos. of standard benchmark functions using CSI-PSO is compared with standard PSO algorithm (Yao *et al.*, 1999) to validate its performance success. The paper is divided into following parts; In part 2, concept of standard PSO is briefly discussed, part- 3 comprises of the concept and formulation of CSI-PSO algorithm. In part- 4, we have shown the 23 benchmark functions used here along with the experimental setting and results of performance of CSI-PSO compared with standard PSO. The last part- 5 encapsulates the contribution of this work and conclusion.

2. Concept of standard PSO

The idea of particle swarm optimization (PSO) is derived by the swarm behavior of birds and fishes. It is a stochastic algorithm based on population just like any other evolutionary algorithms where the swarm particles are made to explore and exploit the search space towards the global solution of optimization problem. Basically the assumption used in PSO is that while every swarm particles exploring and exploiting the search space knows their initial velocities and memorizes their own best local positions and their best global position in the whole swarm group. By using this knowledge every time every swarm starts updating its velocity and position.

On considering a d-dimensional search space with N numbers of swarm particles where the ith particle position attribute is given by $X_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{id})$, the ith particle velocity attribute is given by $vel_i = (vel_{i1}, vel_{i2}, vel_{i3}, ..., vel_{id})$. This velocity attribute indicates the velocity of the particle at the ith position, which is to be updated by using the individual experienced local best position attribute given by $P_{Li} = (P_{Li1}, P_{Li2}, P_{Li3}, ..., P_{Lid})$ and globally the best position is accounted by $Pg = (P_{g1}, P_{g2}, P_{g3}, ..., P_{gd})$. Every swarm particle updates its velocity and position in the search space while moving towards global minimum position is governed by the equations given by

$$vel_{ij} = w * vel_{ij} + \sim_1 * \yen_1 * (P_{Lij} - x_{ij}) + \sim_2 * \yen_2 * (P_{gj} - x_{ij}) \tag{1}$$

$$x_{ij} = x_{ij} + vel_{ij} \tag{2}$$

where j varies from 1 to d and w is the inertia parameter coefficient.
$\sim_1$ and $\sim_2$ are parameter coefficients for acceleration whose values are greater than zero. $\yen_1$ and $\yen_2$ are the randomly generated values between 0 and 1. The individual experience i.e. cognitive term is given by term $\sim_1 * \yen_1 * (P_{Lij} - x_{ij})$ whereas the social term due interaction among the particles is given by term $\sim_2 * \yen_2 * (P_{gj} - x_{ij})$.

The cognitive term helps the particles for exploring the search space while the social term helps the particles for exploiting the search space. From the above equations (1) & (2), it clearly reveals that the swarms are constantly changing their position by changing their instant velocities based upon the information they have about their own earlier position where they were best and the position which is best among the group.

3. Proposed Cognitive and Social Information based PSO (CSI-PSO)

From the last two decades researchers have been greatly motivated by nature and based upon which many algorithms have been proposed. It is observed that with increase in problem size the search space increases. So most of the optimization problems solved by one algorithm may not be solved efficiently in all parameters by others (Wolpert *et al.*, 1997). Hence, there is always an open platform for new heuristic optimization algorithms for solving the optimization problems. In standard PSO, velocity is a vector quantity, where the speed and direction both are involve so it indicates the while updating the velocity the particles not only bring changes in their initial speed at that point of space but also particles inclined in the direction of best position that lead the particles towards global minima. Now here we have assumed that rather than a swarm particle thinks to change its velocity based upon its memory information, if a swarm calculates the position to be changed so that the swarm will reach quickly to its food source because it is very apparent that to reach their destination they made their velocity slow, moderate or fast of their own instinct. So only availing the information about the next position and direction is sufficient enough to reach the destination of global minima. Then the above equation (1) and (2) can be modified as

$$\text{¢}_{ij} = \text{¥}_1(P_{Lij} - x_{ij}) + \text{¥}_2(P_{gj} - x_{ij}) \tag{3}$$

$$x_{ij} = \text{¥}_3 x_{ij} + (1 + \text{¥}_4)\text{¢}_{ij} \tag{4}$$

where $\text{¥}_1, \text{¥}_2, \text{¥}_3, \text{¥}_4$ are randomly generated values between 0 and 1. Equation (3) indicates the probable best calculated destination position the swarm must ahead while Equation (4) indicates the process of updating the position by the swarm particles. The first term of equation (3) indicates the cognitive phenomena while the second term indicates the social information.
The best part proposed CSI-PSO is free from any parameter and moreover the time of convergence is much less compared to standard PSO.

4. Analysis of the results of CSI-PSO

To elucidate the degree of success of our algorithm 23 benchmarks test functions (Yao *et al.*, 1999) were used. All the used benchmark test functions in our experiment are represented in Tables 1-3. Here, Table 1 shows 7 numbers of unimodal test functions, where dim indicates the dimension of the functions. Optimizations methods were specifically designed for unimodal functions to converge, so the rate of convergence these functions put much significant impact for any algorithm performance. Multimodal functions having variable dimension and fixed dimension are shown in Table 2 and Table 3. In multimodal functions more than one local minima were present, so it becomes very difficult for any algorithm to find the global minima. Moreover with increase in dimension the numbers of local minima exponentially increases. The result of optimization for these functions indicates the performance characteristic of the algorithm to avoid being trapped under local minima and how much near the result reach to global minima can be traced by the algorithm.

Table 1 Unimodal benchmark test functions, where dim indicates dimension

Function	Mathematical expression	Range	Global Optima
Sphere Model	$f_1(X) = \sum_{i=1}^{\dim} x_i^2$	$(-100,100)^{\dim}$	0
Schwefel's Problem 2.22	$f_2(X) = \sum_{i=1}^{\dim} \lvert x_i \rvert + \prod_{i=1}^{\dim} \lvert x_i \rvert$	$(-10,10)^{\dim}$	0
Schwefel's Problem 1.2	$f_3(X) = \sum_{i=1}^{\dim} \left(\sum_{j=1}^{i} x_j \right)^2$	$(-100,100)^{\dim}$	0
Schwefel's Problem 2.21	$f_4(X) = \max_i \{ \lvert x_i \rvert, 1 \le i \le n \}$	$(-100,100)^{\dim}$	0
Generalised Rosenbrock's Function	$f_5(X) = \sum_{i=1}^{\dim-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	$(-30,30)^{\dim}$	0

Table 1 (cont'd) Unimodal benchmark test functions, where dim indicates dimension

Function	Mathematical expression	Range	Global Optima
Step Function	$f_6(X)=\sum_{i=1}^{n}\left(\lvert x_i+0.5\rvert\right)^2$	$(-100,100)^{\text{dim}}$	0
Quartic Function i.e. Noise	$f_7(X)=\sum_{i=1}^{\text{dim}} ix_i^4+random[0,1)$	$(-1.28,1.28)^{\text{dim}}$	0

Table 2 Multi-modal test functions, where dim indicates dimension

Function	Mathematical expression	Range	Global Optima
Generalized Schwefel's Problem 2.26	$f_8(X)=\sum_{i=1}^{\text{dim}}-x_i\sin(\sqrt{\lvert x_i\rvert})$	$(-500,500)^{\text{dim}}$	$-418.98*\text{dim}$
Generalized Rastrigin's Function	$f_9(X)=\sum_{i=1}^{\text{dim}}\left[x_i^2-10\cos\left(2\text{®}x_i\right)+10\right]$	$(-5.12,5.12)^{\text{dir}}$	0
Ackley's Function	$f_{10}(X)=-20\exp\left(-0.2\sqrt{\frac{1}{\text{dim}}\sum_{i=1}^{\text{dim}}x_i^2}\right)-\exp\left(\frac{1}{\text{dim}}\sum_{i=1}^{\text{dim}}\cos(2\text{®}x_i)\right)+20+e$	$(-32,32)^{\text{dim}}$	0
Generalized Griewank Function	$f_{11}(X)=\frac{1}{4000}\sum_{i=1}^{\text{dim}}x_i^2-\prod_{i=1}^{\text{dim}}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$(-600,600)^{\text{dim}}$	0
Generalized Penalized Function 1	$f_{12}(X)=\frac{\text{®}}{\text{dim}}\left\{10\sin\left(\text{®}y_1\right)+\sum_{i=1}^{\text{dim}-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\text{®}y_{i+1}\right)\right]+\left(y_{\text{dim}}-1\right)^2\right\}+$ $\sum_{i=1}^{\text{dim}}u\left(x_i,10,100,4\right)$ $where, y_i=1+\frac{x_i+1}{4},\quad u\left(x_i,a,k,m\right)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	$(-50,50)^{\text{dim}}$	0
Generalized Penalized Function 2	$f_{13}(X)=0.1\left\{\sin^2\left(3\text{®}x_1\right)+\sum_{i=1}^{\text{dim}}\left(x_i-1\right)^2[1+\sin^2(3\text{®}x_i+1)]+\left(x_{\text{dim}}-1\right)^2[1+\sin^2(2\text{®}x_{\text{dim}})]\right\}+$ $\sum_{i=1}^{\text{dim}}u\left(x_i,5,100,4\right)$ $where, y_i=1+\frac{x_i+1}{4},\quad u\left(x_i,a,k,m\right)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	$(-50,50)^{\text{dim}}$	0

Table 3 Multi-modal test functions with fixed dimensions

Function	Mathematical expression	Range	Global Optima
Shekel's Foxholes function	$f_{14}(X)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$ $a_{ij}=\begin{bmatrix}-32,-16,0,16,32,-32,...,0,16,32\\-32,-32,-32,-32,-16,...,32,32,32\end{bmatrix}$	$(-65.53,65.53)^2$	$\approx 1\ \ at\ \ (-32,32)$
Kowalik function	$f_{15}(X)=\sum_{i=1}^{11}\left[a_i-x_1\left(b_i^2+b_i x_2\right)/\left(b_i^2+b_i x_3+x_4\right)\right]^2$ $a_i=[0.1957,0.1947,0.1735,0.1600,0.0844,\ldots,$ $0.0627,0.0456,0.0342,0.0323,0.0235,0.0246]$ $b_i^{-1}=[0.25,0.5,1,2,4,6,8,10,12,14,16]$	$(-5,5)^4$	$\approx 0.0003075,$ at $(0.1928,0.1908,0.1231,0.1358)$
Six Hump Camel-Back function	$f_{16}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$(-5,5)^2$	$-1.0316285, at$ $(0.08983,-0.7126)$ $or(0.08983,0.7126)$
Branin Function	$f_{17}(X)=\left(x_2-\frac{5.1}{4\text{®}^2}x_1^2+\frac{5}{\text{®}}x_1-6\right)^2+10\left(1-\frac{1}{8\text{®}}\right)\cos x_1+10$	$(-5,10)\times(0,15)$	$0.398\ \ at(-3.142,12.275)$ $or(3.142,2.275)$ $or(9.425,2.425)$
Goldstein Price Function	$f_{18}(X)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times$ $[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$(-5,5)^2$	$3\ \ at\ \ (0,-1)$
Hartman Family-1	$f_{19}(X)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$ $a=\begin{bmatrix}3&10&30\\0.1&10&35\\3&10&30\\0.1&10&30\end{bmatrix}\quad p=\begin{bmatrix}0.3689&0.117&0.2673\\0.4699&0.4387&0.7470\\0.1091&0.4387&0.7470\\0.0381&0.5743&0.8828\end{bmatrix}$ $c=[1,1,2,3,3,2]$	$(0,1)^3$	$-3.86\ \ at$ $(0.114,0.556,0.852)$
Hartman Family-2	$f_{20}(X)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$ $a=\begin{bmatrix}10&3&17&3.5&1.7&8\\0.05&10&17&0.1&8&14\\3&3.5&17&10&17&8\\17&8&0.05&10&0.1&14\end{bmatrix}$ $p=\begin{bmatrix}0.131&0.169&0.556&0.012&0.828&0.588\\0.232&0.413&0.830&0.373&0.100&0.999\\0.234&0.141&0.352&0.288&0.304&0.665\\0.404&0.882&0.873&0.574&0.109&0.038\end{bmatrix}$ $c=[1,1,2,3,3,2]$	$(0,1)^6$	$-3.32\ \ at$ $(0.201,0.15,0.477,0.275,0.311,0.657)$

Table 3 (cont'd) Multi-modal test functions with fixed dimensions

Function	Mathematical expression	Range	Global Optima
Shekel's Family-1	$f_{21}(X) = -\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T + c_i]^{-1}$ $a = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{bmatrix}$ $c' = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5 \end{bmatrix}$	$(0,10)^4$	-10.1532
Shekel's Family-2	$f_{22}(X) = -\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T + c_i]^{-1}$ a and c values same as f_{21}	$(0,10)^4$	-10.4028
Shekel's Family-3	$f_{23}(X) = -\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T + c_i]^{-1}$ a and c values same as f_{21}	$(0,10)^4$	-10.5363

Here in our experiment we averaged the results over 30 runs and keeping the dimension dim=30 for the functions in the Table 1 and Table 2. The number of iterations was kept 1000. Experimentally setting for standard PSO: $\sim_1 = \sim_2 = 2$ and w varies from 0.9 to 0.2.The results of the simulations are shown in Table 4.

Table 4 Simulation Results

Functions		**Std. PSO**	**CSI-PSO**	**Global Optimum**
1	Achieved Best value	8.033073e-09	**0.000000e+00**	**0**
	Achieved Average	4.635137e-06	**0.000000e+00**	
	Standard Deviation	1.968225e-05	**0.000000e+00**	
	Computing Time per Run	0.768658	**0.546670**	
2	Achieved Best value	5.078186e-06	**0.000000e+00**	**0**
	Achieved Average	4.115557e-04	**0.000000e+00**	
	Standard Deviation	6.560489e-04	**0.000000e+00**	
	Computing Time per Run	0.792497	**0.592173**	
3	Achieved Best value	5.040351e+02	**0.000000e+00**	**0**
	Achieved Average	1.170540e+03	**0.000000e+00**	
	Standard Deviation	5.896480e+02	**0.000000e+00**	
	Computing Time per Run	2.334003	**2.084937**	
4	Achieved Best value	5.252002e+00	**0.000000e+00**	**0**
	Achieved Average	9.461725e+00	**0.000000e+00**	
	Standard Deviation	2.673489e+00	**0.000000e+00**	
	Computing Time per Run	0.795775	**0.543911**	
5	Achieved Best value	**7.457628e+00**	**2.878556e+01**	0
	Achieved Average	1.280729e+04	**2.885955e+01**	
	Standard Deviation	3.083828e+04	**4.881516e-02**	
	Computing Time per Run	0.952693	**0.738134**	

Table 4 (cont'd) Simulation Results

Functions		**Std. PSO**	**CSI-PSO**	**Global Optimum**
6	Achieved Best value	**8.337011e-09**	3.806837e-01	0
	Achieved Average	**1.744402e-05**	9.445605e-01	
	Standard Deviation	**8.590423e-05**	3.303452e-01	
	Computing Time per Run	0.772846	0.542214	
7	Achieved Best value	2.727459e-02	**1.603811e-06**	**0**
	Achieved Average	4.965538e-01	**6.123198e-05**	
	Standard Deviation	1.013664e+00	**5.330837e-05**	
	Computing Time per Run	0.974887	**0.778061**	
8	Achieved Best value	**-1.042698e+04**	-7.716223e+03	-12569.5
	Achieved Average	**-9.081028e+03**	-6.147632e+03	
	Standard Deviation	6.803315e+02	6.829164e+02	
	Computing Time per Run	0.830034	**0.619309**	
9	Achieved Best value	2.089605e+01	**0.000000e+00**	**0**
	Achieved Average	3.969773e+01	**0.000000e+00**	
	Standard Deviation	1.037592e+01	**0.000000e+00**	
	Computing Time per Run	0.865196	**0.596760**	
10	Achieved Best value	2.327036e-05	**8.881784e-16**	**0**
	Achieved Average	5.204727e-01	**8.881784e-16**	
	Standard Deviation	7.522436e-01	**0.000000e+00**	
	Computing Time per Run	0.833737	**0.624138**	
11	Achieved Best value	3.638274e-09	**0.000000e+00**	**0**
	Achieved Average	1.611585e-02	**0.000000e+00**	
	Standard Deviation	2.178763e-02	**0.000000e+00**	
	Computing Time per Run	0.941037	**0.730760**	
12	Achieved Best value	**1.071827e-07**	**5.269191e-02**	0
	Achieved Average	3.279810e-01	**1.478794e-01**	
	Standard Deviation	5.453923e-01	**6.831147e-02**	
	Computing Time per Run	1.642140	1.416239	
13	Achieved Best value	**3.216099e-07**	2.662227e-01	0
	Achieved Average	**8.516446e-03**	9.142214e-01	
	Standard Deviation	**1.130800e-02**	3.523762e-01	
	Computing Time per Run	1.632731	1.399192	
14	Achieved Best value	**9.980038e-01**	**9.980038e-01**	**1**
	Achieved Average	**9.980038e-01**	**9.980397e-01**	
	Standard Deviation	**4.123265e-17**	**1.606160e-04**	
	Computing Time per Run	3.029553	**2.676650**	
15	Achieved Best value	**3.074860e-04**	**3.074788e-04**	**0.0003075**
	Achieved Average	3.552816e-04	**3.348297e-04**	
	Standard Deviation	6.792865e-05	**3.239084e-05**	
	Computing Time per Run	0.839056	**0.823824**	
16	Achieved Best value	**-1.031628e+00**	**-1.031628e+00**	**-1.0316285**
	Achieved Average	**-1.031628e+00**	**-1.031601e+00**	
	Standard Deviation	**6.648565e-16**	**2.559598e-05**	
	Computing Time per Run	0.669986	**0.496887**	
17	Achieved Best value	**3.978874e-01**	**3.978878e-01**	**0.398**
	Achieved Average	**3.978874e-01**	**3.979120e-01**	
	Standard Deviation	**0.000000e+00**	6.375437e-05	
	Computing Time per Run	0.653549	**0.450994**	
18	Achieved Best value	**3.000000e+00**	**3.000000e+00**	**3**
	Achieved Average	**3.000000e+00**	**3.000000e+00**	
	Standard Deviation	**1.068872e-15**	7.412977e-05	
	Computing Time per Run	0.649428	**0.460747**	

Table 4 (cont'd) Simulation Results

Functions		Std. PSO	CSI-PSO	Global Optimum
19	Achieved Best value Achieved Average Standard Deviation Computing Time per Run	-3.859668e+00 -3.452439e+00 2.641498e-01 0.911920	**-3.862679e+00** **-3.862025e+00** **6.134601e-04** **0.689937**	**-3.86**
20	Achieved Best value Achieved Average Standard Deviation Computing Time per Run	-2.946255e+00 -1.838538e+00 6.291126e-01 0.907574	**-3.321345e+00** **-3.274047e+00** **5.949037e-02** **0.729937**	**-3.32**
21	Achieved Best value Achieved Average Standard Deviation Computing Time per Run	-5.588914e+00 -3.610832e+00 8.068221e-01 0.975175	**-1.014021e+01** **-1.007302e+01** **5.976998e-02** **0.750750**	**-10.1532**
22	Achieved Best value Achieved Average Standard Deviation Computing Time per Run	-6.285257e+00 -3.587447e+00 7.635625e-01 1.027297	**-1.039922e+01** **-1.033032e+01** **6.097903e-02** **0.856586**	**-10.4028**
23	Achieved Best value Achieved Average Standard Deviation Computing Time per Run	-6.873460e+00 -4.105762e+00 1.050350e+00 1.258646	**-1.052564e+01** **-1.045940e+01** **7.868030e-02** **0.986546**	**-10.5363**

The graphical comparison results of 23 benchmark functions between CSI-PSO and Standard PSO are shown below.

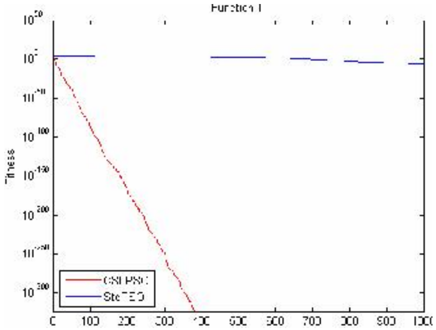

Figure-1- Performance Comparison for CSI-PSO and Std-PSO for function-1

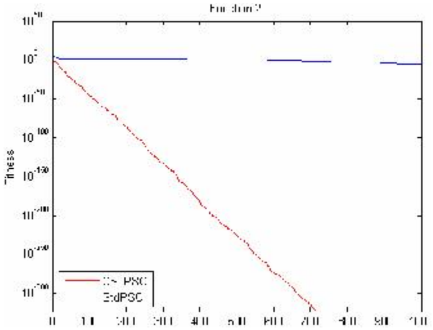

Figure-2- Performance Comparison for CSI-PSO and Std-PSO for function-2

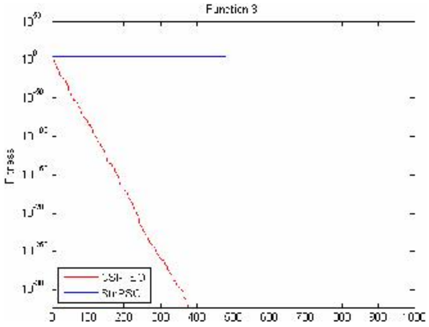

Figure-3- Performance Comparison for CSI-PSO and Std-PSO for function-3

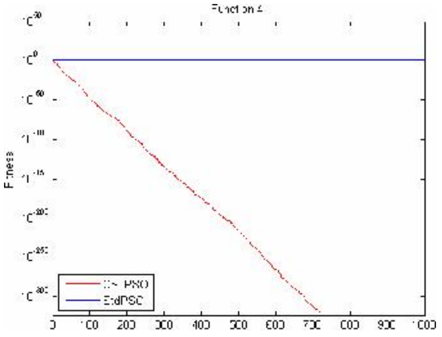

Figure-4- Performance Comparison for CSI-PSO and Std-PSO for function-4

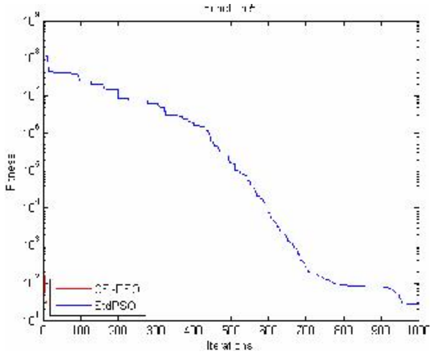

Figure-5- Performance Comparison for CSI-PSO and Std-PSO for function-5

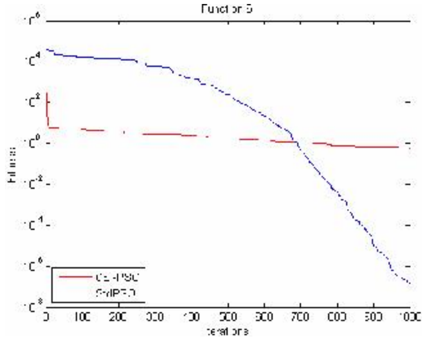

Figure-6- Performance Comparison for CSI-PSO and Std-PSO for function-6

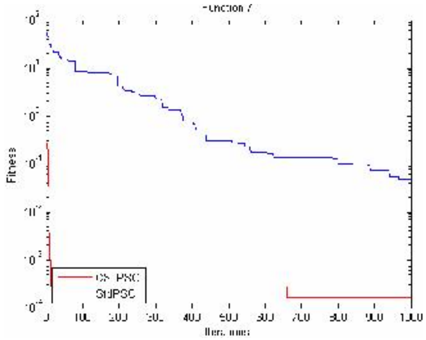

Figure-7- Performance Comparison for CSI-PSO and Std-PSO for function-7

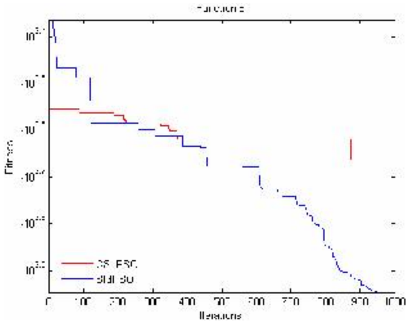

Figure-8- Performance Comparison for CSI-PSO and Std-PSO for function-8

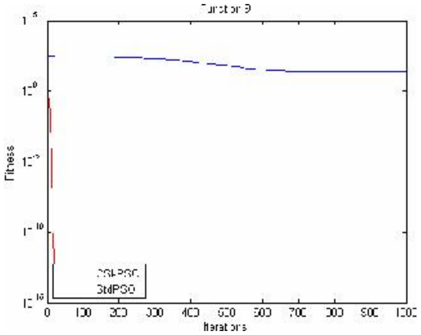

Figure-9- Performance Comparison for CSI-PSO and Std-PSO for function-9

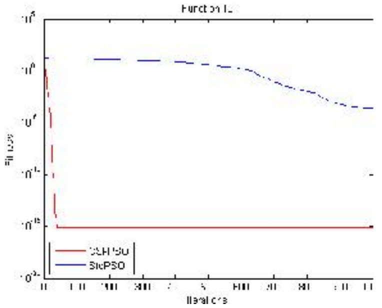

Figure-10- Performance Comparison for CSI-PSO and Std-PSO for function-10

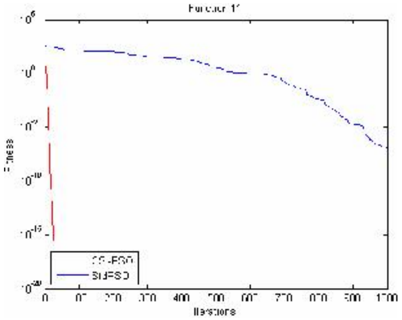

Figure-11- Performance Comparison for CSI-PSO and Std-PSO for function-11

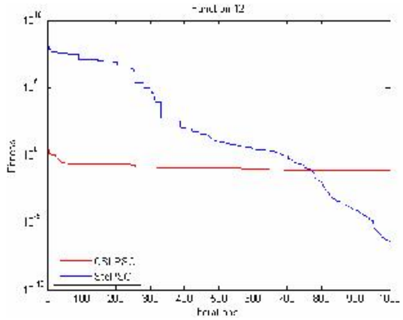

Figure-12- Performance Comparison for CSI-PSO and Std-PSO for function-12

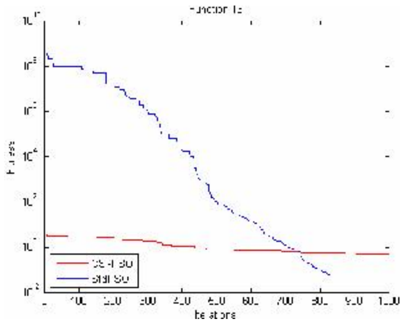

Figure-13- Performance Comparison for CSI-PSO and Std-PSO for function-13

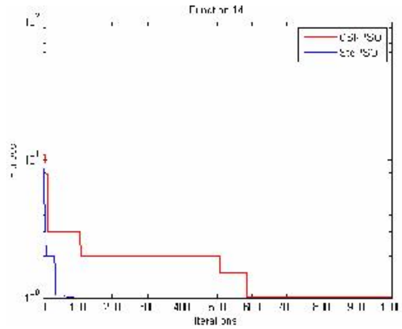

Figure-14- Performance Comparison for CSI-PSO and Std-PSO for function-14

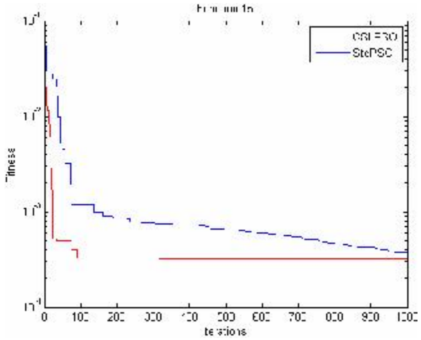

Figure-15- Performance Comparison for CSI-PSO and Std-PSO for function-15

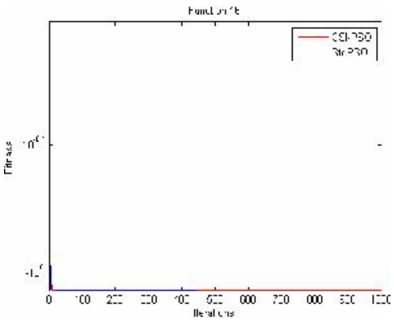

Figure-16- Performance Comparison for CSI-PSO and Std-PSO for function-16

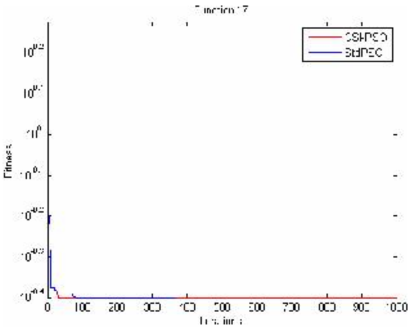

Figure-17- Performance Comparison for CSI-PSO and Std-PSO for function-17

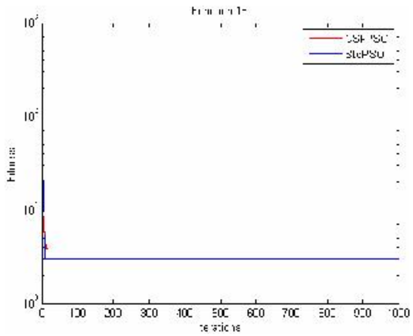

Figure-18- Performance Comparison for CSI-PSO and Std-PSO for function-18

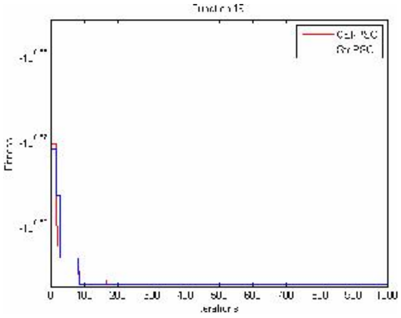

Figure-19- Performance Comparison for CSI-PSO and Std-PSO for function-19

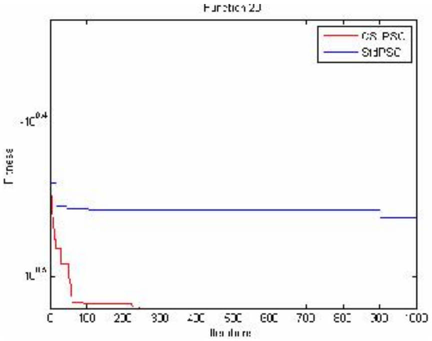

Figure-20- Performance Comparison for CSI-PSO and Std-PSO for function-20

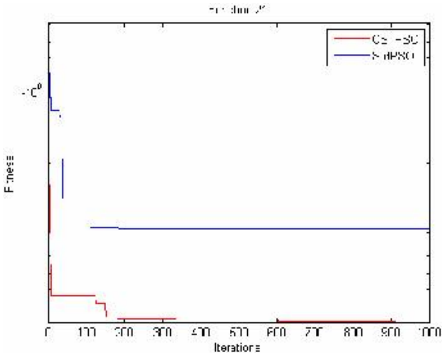

Figure-21- Performance Comparison for CSI-PSO and Std-PSO for function-21

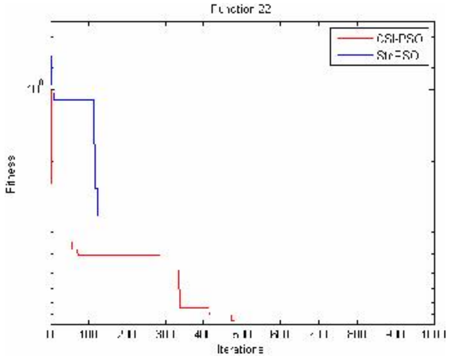

Figure-22- Performance Comparison for CSI-PSO and Std-PSO for function-22

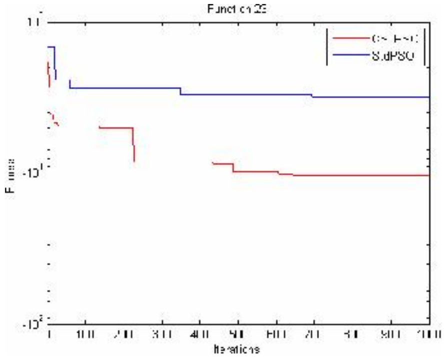

Figure-23- Performance Comparison for CSI-PSO and Std-PSO for function-23

For the evaluation of performance of CSI-PSO, we have calculated the Achieved Best value, Achieved Average of all best, standard deviation and computing time taken per run to converge by the algorithm. For all the functions shown in Table 1-3 the numbers of evaluations done are 1000.Each algorithm is run 30 number of times. The word "Achieved Best value" here stands for the minimum value obtained after running algorithm with every mentioned functions 30 number of times. The word "Achieved Average" implies the mean value after running algorithm for 30 numbers of times. Standard deviation has been calculated among 30 runs of the algorithm. The term 'Computing Time per Run' refers to the average time taken by the algorithm after 1000 numbers of evaluation 30 numbers of times. For the benchmark functions used here lower is the value for all these four parameters of performance evaluation better is the algorithm. Following are the observation made looking at the empirical data obtained and shown in Table 4 and Figure 1 to Figure 23, for the function-1 to function-4, function-7, function-9 to function-11 and function-14 to function-23 the proposed algorithm CSI-PSO has reached or almost touch the global optimum value for the functions defined while taking comparatively less time than standard PSO. But in functions function-5 to function-6, function-8 and function-12 to function-13, standard PSO reflects better results.

5. Conclusion

Every optimization problem has its own characteristic features which should be taken under consideration while optimizing the problem. Indeed in last few years a lot of optimization techniques have been developed but it is not possible by any algorithm to successfully optimize all the optimization problems (Wolpert *et al*. 1997). Almost all optimization algorithms are based upon the hypothetical concepts framed into mathematical expressions to get desired result. Here we have assumed that swam particles only looks into the cognitive and social information. Their initial velocity factor has been neglected keeping the assumption in mind that initial velocity may drag the particles to shift forward to more extent, leads the particles to move away from optimum. The results obtained are very much authenticating our thought and research of renovating and modifying the concept of standard PSO. Here we have limit our research with the concept that only one group of swarm particles are heading towards optimum. Giving a varying thought to swarm concept if assumption has been made that two or more number of swarm groups are heading towards the optimum then apart from social, cognitive information, a new factor neighboring group information may drag the attention of researcher into a another new concept of PSO.

References

Alireza A., 2011, PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems, *Acta Automatica Sinica*, Vol. 37, No.5, pp. 541-549.

Angeline P.J., 1998, Using selection to improve particle swarm optimization, *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, ICEC 1998, Anchorage, AK, USA, pp. 84-89.

Brits R., Engelbrecht A.P., van den Bergh F., 2007, Locating multiple optima using particle swarm optimization, *Applied Mathematics and Computation*, Vol. 189, pp.1859-1883.

Clerc M, Kennedy J. 2002, The particle swarm explosion, stability and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 58-73.

Chatterjee A., Siarry P.. 2006, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, *Computer and Operations Research*, pp. 859-871.

Civicioglu P., 2013, Backtracking search optimization algorithm for numerical optimization problems, *Applied Mathematics and Computation*, Vol. 219, No. 15, pp. 8121-8144.

Eberhart R.C., Kennedy J., 1995, A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.

Eberhart R.C., Shi Y., 1998, Comparison between genetic algorithms and particle swarm optimization. In: Evolutionary programming VII: *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, San Diego CA. Berlin: Springer-Verlag, pp. 611-616.

Kennedy J., Eberhart R.C., 1995, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, Vol. 4, pp. 1942-1948.

Kennedy J., Eberhart R.C., 2001, Swarm Intelligence. Morgan Kaufmann Publishers, Massachusetts, USA.

Liu B, Wang L, Jin YH, Tang F, Huang DX. 2005, Improved particle swarm optimization combined with chaos, *Chaos, Solitons & Fractals*, Vol. 25, pp. 1261-71.

Modares H, Alfi A, Fateh M M. 2010, Parameter identification of chaotic dynamic systems through an improved particle swarm optimization. *Expert Systems with Applications*, Vol. 37, No. 5, pp. 3714-3720.

Nickabadi A., Ebadzadeh M.M., Safabakhsh R., 2011, A novel particle swarm optimization algorithm with adaptive inertia weight, *Applied Soft Computing*, pp.3658-3670.

Sahu A., Panigrahi S.K. & Pattnaik S., 2012, Fast convergence particle swarm optimization for functions optimization, *Procedia Technology*, pp. 319-324.

Shi Y. & Eberhart, R. C., 1998, A modified particle swarm optimizer, *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, 1998, pp. 69-73.

Shi Y., Eberhart R.C., 1999, An empirical study of particle swarm optimization, *Proceedings of the IEEE Congress on Evolutionary Computation*, Piscataway, NJ, pp. 1945-1960.

Shi X.H., Liang Y.C., Lee H.P., Lu C., Wang L.M., 2005, An improved GA and a novel PSO-GA-based hybrid algorithm, *Information Processing Letters*, Vol. 93, No. 5, pp. 255-261.

Suganthan P.N., 1999, Particle swarm optimizer with neighborhood operator, *Proceedings of the 1999 Congress on Evolutionary Computation*, CEC'99, Wasington, DC, USA, pp. 1958-1961

Tripathi P. K., Bandyopadhyay S., Pal S. K. 2007, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Information Sciences*, Vol. 177, pp. 5033-5049.

Van den Bergh F., Engelbrecht A.P., 2006, A study of particle swarm optimization particle trajectories, *Information Sciences*, Vol. 176, pp. 937-971

Wei H.Z., Yang C., Gao L., Yao X., Eberhart R.C., Shi Y., 1998, Extracting rules from fuzzy neural network by particle swarm optimization, *Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage*, Alaska, USA, 1998, pp.74-77.

Wolpert D.H., Macready W.G., 1997, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, 67-82.

Xu X., Pan Z., Xi Y. & Chen L., 2012, Incremental particle swarm optimization, *International Conference on Applied Physics and Industrial Engineering*, pp.1369-1376.

Yao X., Liu Y., Lin G., 1999, Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, Vol. 3, pp. 82-102.

Biographical notes

D. P. Tripathi obtained bachelor's degree in Electronics & Telecommunication Engineering in the year 2002 from Fakir Mohan University, Balasore, India and Completed his M.Tech. in the year 2009 in Communication System Engineering from Veer Surendra Sai University of Technology, Burla, Odisha, India. Having teaching experience of more than 13 years and currently pursuing research in the field of Computer Vision, Image Compression and Soft Computing.

U. R. Jena received B.Sc. Engg. in 1983 from Sambalpur University, Odisha, India, completed M.Tech. in 1997 from I.I.T. Kharagpur, India and Ph.D. in 2003 from Jadavpur university, Kolkata, India. Currently working as Professor in the Department of Electronics and Tele-communication Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India. His research interest includes Computer Vision, Image Processing and Soft Computing. He is having an experience of teaching more than 27 years and also having numbers of publications in both national and international journals and conferences. He is a life time Member of ISTE**.**

Received March 2016
Accepted November 2016
Final acceptance in revised form November 2016