Journal of Artificial Intelligence
ISSN: 2229-3965 & E-ISSN: 2229-3973, Volume 3, Issue 3, 2012, pp.-123-134.
Available online at http://www.bioinfo.in/contents.php?id=71

HPSO: A NEW VERSION OF PARTICLE SWARM OPTIMIZATION ALGORITHM

NARINDER SINGH*, SHARANDEEP SINGH AND SINGH S.B.
Department of Mathematics, Punjabi University, Patiala-147002, Punjab, India.
*Corresponding Author: Email- narindersinghgoria@ymail.com

Received: June 16, 2012; Accepted: September 24, 2012

Abstract- In this paper, a new version of Particle Swarm Optimization Algorithms has been proposed. This algorithm has been developed by combining two different approaches of PSO *i.e.*, Standard Particle Swarm Optimization (SPSO) and Mean Particle Swarm Optimization (MPSO). Numerical experiments for several scalable and non-scalable problems have been done. The results indicate that the proposed algorithm performs better than the existing ones in terms of efficiency, reliability, accuracy and stability

Keywords- SPSO, HPSO (Hybrid Particle Swarm Optimization), global optimization, personal best position, global best position, velocity update equation.

Citation: Narinder Singh, Sharandeep Singh and Singh S.B. (2012) HPSO: A New Version of Particle Swarm Optimization Algorithm. Journal of Artificial Intelligence, ISSN: 2229-3965 & E-ISSN: 2229-3973, Volume 3, Issue 3, pp.-123-134.

Introduction

The particle swarm optimization (PSO) is a new global optimization method based on a metaphor of social interaction [1,2]. Since its inspection PSO is finding applications in all areas of science and engineering [3]. PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other optimization methods.

A PSO algorithm maintains a swarm of individuals (called particles), where each individual (particle) represents a candidate solution. Particles follow a very simple behavior: emulate the success of neighboring particles, and own successes achieved. The position of a particle is therefore influenced by the best particle in a neighborhood, as well as the best solution found by the particle.

Particle position x_i is adjusted using [Equ-1]

$$x_i(k+1) = x_i(k) \oplus v_i(k+1) \quad (1)$$

where the velocity component, $v_i(k)$ represents the step size. For the basic PSO.

$$v_{ij}(k+1) = wv_{ij}(k) \oplus c_1 r_{1j}(y_{ij} - x_{ij}) \oplus c_2 r_{2j}(\hat{y}_j - x_{ij}) \quad (2)$$

The standard Particle Swarm Optimization Algorithm uses the values for inertia weight range = 0.4 to 1.4, acceleration coefficient range = 1.5 to 2.0 respectively, and suggests the upper and lower limits on these values as shown in [Equ-2].

The neighborhood best position y_i, of particle i depends on the neighborhood topology used [4].

Shi and Eberhart [11] proposed to use an "inertia weight" parameter [Equ-3].

$$v_{ij}(k+1) = w \otimes v_{ij}(k) \oplus c_1 r_{1j} \otimes (y_{ij} - x_{ij}) \oplus c_2 r_{2j} \otimes (\hat{y}_j - x_{ij}) \quad (3)$$

Eberhart and Shi suggested to use the inertia weight which decreasing over time, typically from 0.9 to 0.4. It has the effect of narrowing the search, gradually changing from an exploratory to an exploitative mode.

Clerc and Kennedy [34] suggested a more generalized PSO, where a constriction coefficient is applied to both terms of the velocity formula. The authors showed that the constriction PSO can converge without using [Eua-4].

Vmax: $v_{ij}(k+1) = \chi \otimes (v_{ij}(k) \oplus c_1 r_{1j} \otimes (y_{ij} - x_{ij}) \oplus c_2 r_{2j} \otimes (\hat{y}_j - x_{ij}))$ (4)

where the constriction factor was set 0.7289. Clerc and Kennedy [33]. By using the constriction coefficient, the amplitude of the particle's oscillation decreases, resulting in its convergence over time.

PSO variants are continually being devised in an attempt to overcome this deficiency (see e.g. [18-26] for a few recent additions). These PSO variants greatly increase the complexity of the original methods. Pedersen and co workers [27,28] have demonstrated that satisfactory performance can be achieved with the basic PSO if only its parameters are properly tuned.

Optimization

Optimization is the mechanism by which one finds the maximum or minimum value of a function or process. This mechanism is used in fields such as Physics, Chemistry, Economics and Engineering where the goal is to maximize efficiency, production, or some other measure. Optimization can refer to either minimization or maximization; maximization of a function f is equivalent to minimization of the opposite of this function.

Mathematically, a minimization task is defined as:

$$\text{Given } f : R^n \to R$$
$$\text{Find } \hat{x} \in R^n \text{ such that } f(\hat{x}) \le f(x), \quad \forall x \in R^n$$

Similarly, a maximization task is defined as:

$$\text{Given } f : R^n \to R$$
$$\text{Find } \hat{x} \in R^n \text{ such that } f(\hat{x}) \ge f(x), \quad \forall x \in R^n$$

The domain R^n of f is referred to as the search space (or parameter space). Each element of R^n is called a candidate solution in the search space, with $\hat{x}$ being the optimal solution. The value n denotes the number of dimensions of the search space, and thus the number of parameters involved in the optimization problem. The function f is called the objective function, which maps the search space to the function space. Since a function has only one output, this function space is usually one-dimensional. The function space is then mapped to the one-dimensional fitness space, providing a single fitness value for each set of parameters.

Related Work

Tang Ziyu and Zhang Dingxue [16] proposed a new version of PSO without the velocity of the previous iteration, and a novel selection of acceleration coefficients was introduced in the algorithm. To overcome premature searching for the velocity of a particle at zero, reinitialize the velocity of the particle with a random velocity. Simultaneity, to enhance exploration in the early stage and exploitation during the latter, we introduce exponential time-varying acceleration coefficients. The simulation results show that the algorithm has better probability of finding global optimum and mean best value than others algorithm.

Particle swarm optimization is a very useful EA related technique [29-31], with various variants viz [32]: 2-D Otsu PSO (TOPSO), Active Target PSO (APSO), Adaptive PSO (APSO), Adaptive Mutation PSO (AMPSO), Adaptive PSO Guided by Acceleration Information (AGPSO), Angle Modulated PSO (AMPSO), Attractive Repulsive Particle Swarm Optimization (ARPSO), Augmented Lagrangian PSO (ALPSO), Best Rotation PSO (BRPSO), Binary PSO (BPSO), Co-evolutionary PSO, Combinatorial PSO (CPSO), Comprehensive Learning PSO (CLPSO), Concurrent PSO (CONPSO), Constrained optimization via PSO (COPSO), Cooperative PSO (CPSO_M), Cooperative PSO (CPSO_S), Cooperatively Coevolving Particle Swarms (CCPSO), Cooperative Multiple PSO (CMPSO), Cultural based PSO (CBPSO), Dissipative PSO (DPSO), Divided range PSO (DRPSO), Dual Similar PSO Algorithm (DSPSOA), Dynamic adaptive dissipative PSO (ADPSO), Dynamic and Adjustable PSO (DAPSO), Dynamic Double Particle Swarm Optimizer (DDPSO), Dual Layered PSO (DLPSO), Dynamic neighborhood PSO (DNPSO), Estimation of Distribution PSO (EDPSO), Evolutionary Iteration PSO (EIPSO), Evolutionary Programming and PSO (EPPSO), Extended Particle Swarms (XPSO), Extended PSO (EPSO), Fitness-to-Distance Ratio PSO (FDRPSO) and so on and so forth.

The concept of PSO was originally developed by Kennedy J. and Eberhart R.C. [2] using Particle Swarm Optimization methodology.

Kusum Deep and Bansal [35] developed a new approach by replacing two terms of original velocity update equation by two new terms based on the linear combination of pbest and gbest. Its performance is compared with the Standard PSO (SPSO) by testing it on a set of 15 scalable and 15 non-scalable test problems. Based on the numerical and graphical analyses of results it was concluded that the MeanPSO outperforms the SPSO, in terms of efficiency, reliability, accuracy and stability

Singh and Singh [17] derived a new OHGBPPSO. When the particle size is increasing and decreasing in the swarm, the proposed algorithm outperforms the Standard Particle Swarm Optimization. But in the case when the particle size is fixed and no particle enters/leaves the swarm the Standard Particle Swarm Algorithm is better than the proposed one.

The Proposed Algorithm

The motivation behind designing HPSO is to accelerate its rate of convergence and rate of success in finding global optimal solution. In new approach in the velocity update equation we have summed up velocity update equation of Standard Particle Swarm Optimization and Mean Particle Swarm Optimization. Thus, we have introduced a new velocity update equation as follows [Equ-5]:

$$v_{ij}(k+1) = wv_{ij}(k) \oplus c_1 r_{1j}(y_{ij} - x_{ij}) \oplus c_2 r_{2j}(\hat{y}_j - x_{ij}) \oplus$$
$$wv_{ij}(k) \oplus c_1 r_{1j}\left(\frac{(y_{ij} \oplus \hat{y}_j)}{2} - x_{ij}\right) \oplus c_2 r_{2j}\left(\frac{(y_{ij} - \hat{y}_j)}{2} - x_{ij}\right)$$

i.e.

$$v_{ij}(k+1) = 2wv_{ij}(k) \oplus c_1 r_{1j}\left(\frac{(3y_{ij} \oplus \hat{y}_j)}{2} - 2x_{ij}\right) \oplus c_2 r_{2j}\left(\frac{(y_{ij} \oplus \hat{y}_j)}{2} - 2x_{ij}\right) \quad (5)$$

In the velocity update equation of this new PSO the first term represents the current velocity of the particle and can be thought of as a momentum term. The second term is proportional to the vector $c_1 r_{1j}\left(\frac{(3y_{ij} \oplus \hat{y}_j)}{2} - 2x_{ij}\right)$, is responsible for the attractor of particle's current position and positive direction of its own best position (pbest). The third term is proportional to the vector $c_2 r_{2j}\left(\frac{(y_{ij} \oplus \hat{y}_j)}{2} - 2x_{ij}\right)$, is responsible for the attractor of particle's current position.

The pseudo code of HPSO is shown below:

Algorithm- HPSO

The pseudo code of the procedure is as follows [Fig-1]:

For each particle

Initialize particle

END

Do

For each particle

Calculate fitness value

If the fitness value is better than the best fitness value pbest (personal best position) in history, set current value as the new pbest

End

Choose the particle with the best fitness value of all the particles as the gbest (global best position)

For each particle

Calculate particle velocity according equation

$$v_{ij}(k+1)=2wv_{ij}(k)\oplus c_1 r_{1j}\left(\frac{(3y_{ij}\oplus\hat{y}_j)}{2}-2x_{ij}\right)\oplus c_2 r_{2j}\left(\frac{(y_{ij}\oplus\hat{y}_j)}{2}-2x_{ij}\right)$$

Update particle position according equation

$$x_i(k+1)=x_i(k)\oplus v_i(k+1)$$

End

While maximum iterations or minimum error criteria is not attained.

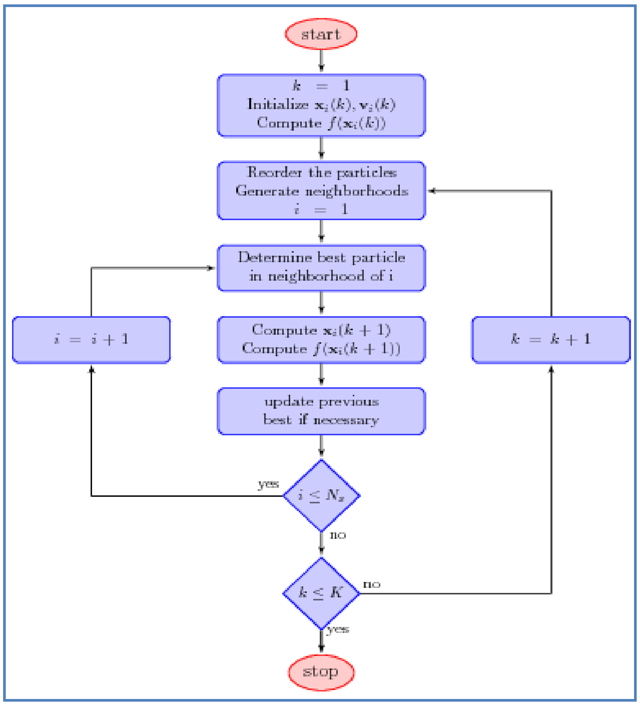

Fig. 1- Flow Chart of HPSO

[Fig-2] reflects comparison between the particle position in SPSO and HPSO Algorithm.

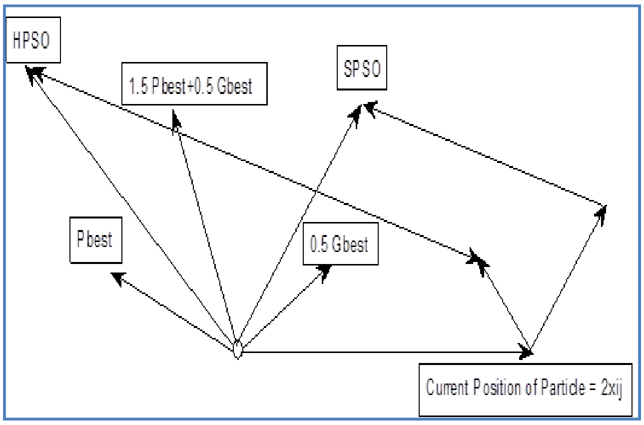

Fig. 2- Comparison of SPSO Particle and HPSO Particle by Scalable Problems

Parameter Setting

Computational Experiments were performed to fine tune the values of various parameters for its best performance. For that purpose all measure values of parameters viz. inertia weight in the range [0.4, 0.9] and acceleration coefficient in the range [1.5, 2.0] were tested.

Test Problems

Many times it is found that the evaluation of a proposed algorithm is evaluated only on a few scalable and non scalable problems. However, in this paper we consider a test of 15 scalable and 13 non scalable problems with varying difficulty levels and problem size. The performance of Standard Particle Swarm Optimization Algorithm and newly proposed HPSO has been verified on these two types of problem sets.

Detail of 15 Scalable Problems SET-I (Continued)

Problem I (Ackley):

$$Min\ f(x)=-20\exp\left(-0.02\sqrt{n^{-1}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(n^{-1}\sum_{i=1}^{n}\cos(\pi x_i)\right)+20+e$$

In which search space lies between $-30 \le x_i \le 30$ and Minimize Objective Function Value is 0.

Problem II (Cosine Mixture):

$$Min\ f(x)=-0.1\sum_{i=1}^{n}\cos(5\pi x_i)+\sum_{i=1}^{n}x_i^2$$

In which search space lies between $-1 \le x_i \le 1$ and Minimize Objective Function Value is $-0.1\times(n)$.

Problem III (Exponential):

$$Min\ f(x)=(-0.5\sum_{i=1}^{n}x_i^2)$$

In which search space lies between $-1 \le x_i \le 1$ and Minimize Objective Function Value is -1.

Problem IV (Griewank):

$$Min\, f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$$

In which search space lies between $-600 \le x_i \le 600$ and Minimize Objective Function Value is 0.

Problem V (Rastrigin):

$$Min\, f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$$

In which search space lies between $-5.12 \le x_i \le 5.12$ and Minimize Objective Function Value is 0.

Problem VI (Function '6'):

$$Min\, f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

In which search space lies between $-30 \le x_i \le 30$ and Minimize Objective Function Value is 0.

Problem VII (Zakharov's):

$$Min\, f(x) = \sum_{i=1}^{n} x_i^2 + [\sum_{i=1}^{n} (\frac{i}{2})x_i]^2 + [\sum_{i=1}^{n} (\frac{i}{2})x_i]^4$$

In which search space lies between $-5.12 \le x_i \le 5.12$ and Minimize Objective Function Value is 0.

Problem VIII (Sphere):

$$Min\, f(x) = \sum_{i=1}^{n} x_i^2$$

In which search space lies between $-5.12 \le x_i \le 5.12$ and Minimize Objective Function Value is 0.

Problem IX (Axis Parallel Hyper Ellipsoid):

$$Min\, f(x) = \sum_{i=1}^{n} ix_i^2$$

In which search space lies between $-5.12 \le x_i \le 5.12$ and Minimize Objective Function Value is 0.

Problem X (Schwefel '3'):

$$Min\, f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Problem XI (Dejong):

$$Min\, f(x) = \sum_{i=1}^{n} (x_i^4 + rand(0,1))$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Problem XII (Schwefel '4'):

$$Min\, f(x) = Max\{|x_i|, 1 \le i \le n\}$$

In which search space lies between $-100 \le x_i \le 100$ and Minimize Objective Function Value is 0.

Problem XIII (Cigar):

$$Min\, f(x) = x_i^2 + 100000 \sum_{i=1}^{n} x_i^2$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Problem XIV (Brown '3'):

$$Min\, f(x) = \sum_{i=1}^{n-1} [(x_i^2)(x_{i+1}^2 + 1) + (x_{i+1}^2 + 1)(x_i^2 + 1)]$$

In which search space lies between $-1 \le x_i \le 4$ and Minimize Objective Function Value is 0.

Problem XV (Function '15'):

$$Min\, f(x) = \sum_{i=1}^{n} ix_i^2$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Detail of 13 Non- Scalable Problems SET-II

Problem I (Becker and Lago):

$$Min\, f(x) = (|x_1| - 5)^2 + (|x_2| - 5)^2$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Problem II (Bohachevsky '1'):

$$Min\, f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

In which search space lies between $-50 \le x_i \le 50$ and Minimize Objective Function Value is 0.

Problem III (Bohachevsky '2'):

$$Min\ f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$$

In which search space lies between $-50 \le x_i \le 50$ and Minimize Objective Function Value is 0.

Problem IV (Branin):

$$Min\ f(x) = a(x_2 - bx_1^2 + cx_1 - d)^2 + g(1-h)\cos(x_1) + g$$

$$a = 1,\ b = \frac{5.1}{4\pi^2},\ c = \frac{5}{\pi},\ d = 6,\ g = 10,\ h = \frac{1}{8\pi}$$

In which search space lies between $-5 \le x_1 \le 100$, $-5 \le x_2 \le 15$ and Minimize Objective Function Value is 0.398.

Problem V (Eggcrate):

$$Min\ f(x) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$$

In which search space lies between $-2\pi \le x_i \le 2\pi$ and Minimize Objective Function Value is 0.

Problem VI (Miele and Cantrell):

$$Min\ f(x) = (\exp(x_1) - x_4)^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8$$

In which search space lies between $-1 \le x_i \le 1$ and Minimize Objective Function Value is 0.

Problem VII (Modified Rosenbrock):

$$Min\ f(x) = 100(x_2 - x_1^2)^2 + [6.4(x_2 - 0.5)^2 - x_1 - 0.6]^2$$

In which search space lies between $-5 \le x_1, x_2 \le 5$ and Minimize Objective Function Value is 0.

Problem VIII (Easom):

$$Min\ f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is -1

Problem IX (Periodic):

$$Min\ f(x) = -1 + \sin^2 x_1 + \sin^2 x_2 - 0.1\exp(-x_1^2 - x_2^2)$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.9.

Problem X (Powell's):

$$Min\ f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is 0.

Problem XI (Camel Back-3):

$$Min\ f(x) = 2x_1^2 + 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2$$

In which search space lies between $-5 \le x_1, x_2 \le 5$ and Minimize Objective Function Value is 0.

Problem XII (Camel Back-6):

$$Min\ f(x) = 4x_1^2 + 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

In which search space lies between $-5 \le x_1, x_2 \le 5$ and Minimize Objective Function Value is -1.0316.

Problem XIII (Aluffi-Pentini's):

$$Min\ f(x) = 0.25x_1^4 - 0.5x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2$$

In which search space lies between $-10 \le x_i \le 10$ and Minimize Objective Function Value is -0.352.

Analysis

In SPSO and new proposed algorithm HPSO the balance between the local and global exploration abilities is mainly controlled by the inertia weight. The numerical experimental results have been performed to illustrate this. By setting the maximum velocity allowed to be two, it was found that PSO with an inertia weight in the range [0.4, 0.9] on average has a better performance; that is, it has a large chance to find the global optimum within a reasonable number of iterations.

A number of criteria has been used to evaluate the performance of SPSO with HPSO. The percentage of success is used to evaluate the reliability. The average number of function evaluations of successful runs and the average computational time of the successful runs, are used to evaluate the cost. For Problem Set 1, the quality of the solution obtained is measured by the minimum, mean and standard deviation of the objective function values out of thirty runs. This is shown in [Table-1], [Table-3], [Table-5], [Table-7] and [Table-9]. The corresponding information for Problem Set 2 is shown in [Table-2], [Table-4], [Table-6], [Table-8] and [Table-10] respectively.

Firstly, we are testing the SPSO and new approach HPSO on the parameter setting as: swarm size 30 dim, function evaluation 30,000, inertia weight 0.5 and acceleration coefficient 1.3. The result of [Table-1] and [Table-2], it can be shown that SPSO gives a better quality of solutions as compared to HPSO. But on the setting this parameter SPSO and HPSO cannot solve 100% success of rate of the two scalable and four non-scalable problems.

Secondly, we are testing the SPSO and HPSO on the parameter setting as: swarm size 30 dim, function evaluation 30,000, inertia weight 0.6, 0.8 and acceleration coefficient 1.4, 1.6. From the results of [Table-3], [Table-4], [Table-5] and [Table-6], it is concluded that SPSO and HPSO could not solve two scalable and five non-scalable problems with 100% success.

Table1- Comparison of minimum objective function value of SPSO and HPSO for 15 Scalable Problem Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	0.387931	2.213546	0.463183	2.785904	0.028413	0.154321	100%	0.00%
2	1	1	1	1	0	0	0.00%	0.00%
3	0.080679	0.098733	0.127769	0.138034	0.023686	0.026596	100%	100%
4	7.959672	40.31155	22.20173	52.49853	3.815774	5.94983	0.00%	0.00%
5	0.372065	2.083941	0.451844	2.722288	0.035701	0.32848	100%	0.00%
6	0.000011	0.000016	0.045324	0.093635	0.059453	0.116726	100%	100%
7	0.000002	0.000013	0.034576	0.054799	0.061729	0.085849	100%	100%
8	0	0	0.000221	0.000259	0.000494	0.000406	100%	100%
9	0	0.000002	0.00641	0.007523	0.01432	0.011785	100%	100%
10	0.000183	0.000489	0.020887	0.024727	0.021163	0.020645	100%	100%
11	0.204376	4.340918	0.399203	8.632978	0.070762	2.960194	100%	100%
12	0.000092	0.000244	0.010444	0.012363	0.010581	0.010322	100%	100%
13	0.000873	0.158666	0.238075	13.71884	0.155831	10.9795	100%	4.00%
14	0.000357	0.000506	0.023792	0.04026	0.0215	0.036367	100%	100%
15	0	0	0.000052	0.000064	0.000103	0.000127	100%	100%

Analysis	SPSO	HPSO
Swarm Size	30 dim	30 dim
Function Evaluation	30,000	30,000
Inertia Weight	0.5	0.5
Acceleration Coefficient	1.3	1.3

Table 2- Comparison of minimum objective function value of SPSO and HPSO for 13 Non-Scalable Problem Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	25	25	25	25	0	0	0.00%	0.00%
2	0.000252	0.000485	0.014243	0.018005	0.025041	0.025188	100%	100%
3	0.000274	0.000274	0.014718	0.017547	0.01185	0.017243	100%	100%
4	27.71683	27.71683	27.71683	27.71683	0	0	0.00%	0.00%
5	0.00045	0.003603	0.189215	0.219694	0.130739	0.142171	100%	100%
6	73046.6	85046.6	73046.6	89046.6	0	0	0.00%	0.00%
7	20.70113	217.384	22.42282	339.9513	1.234575	58.71676	0.00%	0.00%
8	0	0.00002	0	0	0	0	100%	100%
9	0.580465	1.380465	2.780465	1.380465	0	0	100%	0.00%
10	0.002558	0.025302	0.224278	0.26239	0.133917	0.136451	100%	100%
11	0.000096	0.000049	0.008199	0.017657	0.007526	0.016862	100%	100%
12	0.000004	0.000014	0.000789	0.010052	0.005954	0.008192	100%	100%
13	0.000004	0.000008	0.001258	0.001605	0.001139	0.001605	100%	100%

Analysis	SPSO	HPSO
Swarm Size	30 dim	30 dim
Function Evaluation	30,000	30,000
Inertia Weight	0.6	0.6
Acceleration Coefficient	1.4	1.4

Table 3- Comparison of minimum objective function value of SPSO and HPSO for 15 Scalable Problem Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	0.39522	2.82538	0.469694	3.094737	0.027105	0.123042	100%	0.00%
2	1	1	1	1	0	0	0.00%	0.00%
3	0.094588	0.101718	0.148249	0.188355	0.02232	0.034292	100%	100%
4	16.92943	33.20485	25.94843	45.77758	2.620415	5.845835	0.00%	0.00%
5	0.335672	3.011404	0.442806	3.957319	0.045047	0.439678	100%	0.00%
6	0	0.000044	0.06982	0.081741	0.117727	0.10061	100%	100%
7	0	0.000083	0.041217	0.064499	0.057818	0.076679	100%	100%
8	0	0	0.000195	0.000305	0.000274	0.000363	100%	100%
9	0	0.000011	0.005658	0.008854	0.007937	0.010526	100%	100%
10	0.000073	0.001256	0.021234	0.028744	0.018154	0.019877	100%	100%
11	0.259205	6.941149	0.420516	17.56031	0.057058	5.788724	100%	0.00%
12	0.000037	0.000628	0.010617	0.014372	0.009077	0.009938	100%	100%
13	0.004746	0.083234	0.253189	15.18506	0.11678	14.23525	100%	2.00%
14	0.000629	0.000658	0.035198	0.040451	0.03469	0.040122	100%	100%
15	0	0	0.000034	0.000074	0.000049	0.00012	100%	100%

Table 4- Comparison of minimum objective function value of SPSO and HPSO for 13 Non-Scalable Problem Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	25	25	25	25	0	0	0.00%	0.00%
2	0.000185	0.000211	0.010405	0.013367	0.021601	0.025188	100%	100%
3	0.000274	0.000274	0.014355	0.016635	0.010956	0.012941	100%	100%
4	27.71683	27.71683	27.71683	27.71683	0	0	0.00%	0.00%
5	0.003603	0.006597	0.197137	0.214412	0.14323	0.145243	100%	100%
6	73046.6	73046.6	73046.6	73046.6	0	0.000001	0.00%	0.00%
7	22.41474	356.7507	23.55341	506.359	0.342964	93.16889	0.00%	0.00%
8	0	0	0	0	0	0	100%	100%
9	1.380465	1.380465	1.380465	1.380465	0	0	0.00%	0.00%
10	0.010489	0.005826	0.242355	0.287753	0.12514	0.139614	100%	100%
11	0.000388	0.000186	0.011989	0.022206	0.010398	0.021582	100%	100%
12	0.000019	0.00002	0.007385	0.012785	0.00758	0.014302	100%	100%
13	0.000016	0.000016	0.001705	0.002644	0.001511	0.002192	100%	100%

Analysis	SPSO	HPSO
Swarm Size	30 dim	30 dim
Function Evaluation	30,000	30,000
Inertia Weight	0.8	0.8
Acceleration Coefficient	1.6	1.6

Table 5- Comparison of minimum objective function value of SPSO and HPSO for 15 Scalable Problem Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	0.302491	2.908172	0.691205	3.262553	0.32316	0.087067	62.00%	0.00%
2	1	1	1	1	0	0	0.00%	0.00%
3	0.059008	0.192762	0.158936	0.27673	0.026272	0.03494	100%	100%
4	25.93121	29.30517	27.37303	37.7589	0.966569	4.39901	0.00%	0.00%
5	0.410389	4.720973	0.473893	6.463293	0.02743	0.708598	100%	0.00%
6	0.000008	0.000017	0.071594	0.109527	0.107192	0.117171	100%	100%
7	0.000031	0.000448	0.042033	0.097578	0.062633	0.12661	100%	100%
8	0	0	0.000199	0.000814	0.000296	0.001687	100%	100%
9	0.000004	0.000005	0.00577	0.023604	0.008598	0.048929	100%	100%
10	0.000763	0.000807	0.021675	0.040952	0.018058	0.039732	100%	100%
11	0.320697	24.19404	0.467348	44.75876	0.045955	9.381633	100%	0.00%
12	0.000382	0.000404	0.010838	0.020476	0.009029	0.019866	100%	100%
13	0.009936	0.061379	0.275779	28.77909	0.132544	29.07476	100%	2.00%
14	0.000407	0.000658	0.031531	0.041339	0.031074	0.034328	100%	100%
15	0	0	0.000039	0.000157	0.000052	0.000319	100%	100%

Table 6- Comparison of minimum objective function value of SPSO and HPSO for 13 Non-Scalable Problem Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	25	25	25	25	0	0	0.00%	0.00%
2	0.000485	0.000485	0.012163	0.021492	0.009654	0.033967	100%	100%
3	0.000274	0.000274	0.012683	0.016529	0.008866	0.013177	100%	100%
4	27.71683	27.71683	27.71683	27.71683	0	0	0.00%	0.00%
5	0.000973	0.000788	0.134761	0.204412	0.124735	0.13335	100%	100%
6	73046.6	73046.6	73046.6	73046.6	0	0.000001	0.00%	0.00%
7	28.35727	358.851	37.83285	760.6364	11.33987	136.7744	0.00%	0.00%
8	0	0	0	0	0	0	100%	100%
9	1.380465	1.380465	1.380465	1.380465	0	0	0.00%	0.00%
10	0.013064	0.051677	0.239829	0.358124	0.131305	0.152438	100%	90.00%
11	0.000025	0.000348	0.012279	0.031316	0.011362	0.028257	100%	100%
12	0.000028	0.000971	0.008485	0.015242	0.01002	0.013412	100%	100%
13	0.000016	0.000016	0.00171	0.003068	0.001651	0.002486	100%	100%

Analysis	SPSO	HPSO
Swarm Size	30 dim	30 dim
Function Evaluation	30,000	30,000
Inertia Weight	0.9	0.9
Acceleration Coefficient	1.7	1.7

Thirdly, we are testing the SPSO and HPSO on the parameter setting as swarm size 30 dim, function evaluation 30,000, inertia weight 0.9 and acceleration coefficient 1.7. From the results of] [Table-7] and [Table-8], it is concluded SPSO and HPSO could not solve four scalable and four non-scalable problems.

Table 7- Comparison of minimum objective function value of SPSO and HPSO for 15 Scalable Problem Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	2.09691	3.148713	2.709167	3.307034	0.209419	0.05205	62.00%	0.00%
2	1	1	1	1	0	0	0.00%	0.00%
3	0.063837	0.192762	0.160985	0.291546	0.029135	0.034786	100%	100%
4	27.00595	27.79408	27.78788	35.05702	0.609905	4.018404	0.00%	0.00%
5	1.536115	4.563332	2.630977	6.749619	0.568664	0.719269	0.00%	0.00%
6	0	0.000016	0.066672	0.113697	0.092314	0.113536	100%	100%
7	0.00002	0	0.039573	0.077499	0.048887	0.115932	100%	100%
8	0	0	0.000187	0.000877	0.000231	0.001838	100%	100%
9	0.000003	0	0.005432	0.025444	0.006711	0.053295	100%	100%
10	0.000619	0.00004	0.022017	0.04167	0.016266	0.042109	100%	100%
11	2.089925	22.46909	6.810866	46.77206	2.541166	8.735867	0.00%	0.00%
12	0.000309	0.00002	0.011008	0.020835	0.008133	0.021055	100%	100%
13	0.004454	0.884075	0.215886	39.08755	0.134588	36.85742	100%	0.00%
14	0.000382	0.000658	0.033447	0.043317	0.028908	0.037955	100%	100%
15	0	0	0.000047	0.000186	0.000069	0.000393	100%	100%

Table 8- Comparison of minimum objective function value of SPSO and HPSO for 13 Non-Scalable Problem Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	25	25	25	25	0	0	0.00%	0.00%
2	0.000088	0.000485	0.011417	0.023255	0.008292	0.027897	100%	100%
3	0.000274	0.000274	0.012265	0.022954	0.006836	0.038757	100%	100%
4	27.71683	27.71683	27.71683	27.71683	0	0	0.00%	0.00%
5	0.002251	0.002542	0.1857	0.236147	0.146323	0.148277	100%	100%
6	73046.6	73046.6	73046.6	73046.6	0	0.000002	0.00%	0.00%
7	0.14845	499.5557	227.9079	784.9403	139.8914	129.3896	8.00%	0.00%
8	0	0	0	0	0	0	100%	100%
9	1.380465	1.380465	1.380465	1.380465	0	0	0.00%	0.00%
10	0.030838	0.011159	0.251399	0.223675	0.12742	0.376073	100%	72.00%
11	0.000352	0.001827	0.014684	0.037675	0.014275	0.03201	100%	100%
12	0.00008	0.000006	0.009637	0.015294	0.010168	0.014321	100%	100%
13	0.000016	0.000016	0.001704	0.002428	0.001381	0.001928	100%	100%

Continuing in the same manner authors concluded that the parameter setting of three weight factors w, c_1 and c_2 at 0.7, 1.5, 1.5, swarm size = 30 and function evaluation = 30,000 respectively provides the best convergence rate for the scalable and non-scalable problems considered. Other combination of parameter values usually lead to much slower convergence or sometimes non -convergence at all.

Experiments and Discussion on the Results

Performance of the algorithm was tested on a set of 28 benchmark Problems (15 Scalable and 13 Non-Scalable). The scalable and non-scalable problems were chosen as the test problems. The Standard Particle Swarm Optimization implementation was written in C and compiled using the Borland C++ Version 4.5 compiler. For the purpose of comparison, all the simulation use the parameter setting of the SPSO implementation except the inertia weight w, acceleration coefficient, swarm size and maximum velocity allowed. The swarm size (number of particles) is 30. The dynamic range for each element of a particle is defined as (-100, 100), that is, the particle cannot move out of this range in each dim and thus Xmax = 100. The maximum number of iterations allowed is 30,000. If the SPSO and HPSO implementation cannot find an acceptable solution within 30,000 iterations, it is ruled that it fails to find the global optimum in this run.

As stated earlier in section 7, the parameter setting the three weight factors w, c_1 and c_2 at 0.7, 1.5, 1.5, swarm size = 30 and function evaluation = 30,000 respectively provides the best convergence rate for the scalable and non-scalable problems considered. Other combination of parameter values usually lead to much slower convergence or sometimes non convergence at all.

In observing [Table-9], the quality of the solution obtained is measured by the minimum function value, mean function value, standard deviation and success of rate, the objective function values out of 30 runs. it can be seen that HPSO gives a better quality of solutions as compared to SPSO. Thus, for the scalable problems HPSO outperforms SPSO with respect to efficiency, reliability, cost and robustness. From the [Table-1], it can be shown that the new algorithm HPSO solve all the scalable problem with 100% success while SPSO cannot solve all the scalable problems 100% successfully.

Table 9- Comparison of minimum objective function value of SPSO and HPSO for 15 Scalable Problem Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	0.667619	0.438935	16485.6	2016	0.142795	0.115137	98.00%	100%
2	0.644392	0.403938	1708.2	174.6	0.053545	0.133075	100%	100%
3	0	0	60	60	0.000207	0.000282	100%	100%
4	0.777974	0.356199	14364.6	3393.6	0.026005	0.12876	100%	100%
5	27.12782	0.244824	30000	15957	29.80959	14.36265	0.00%	100%
6	0.000061	0.000037	166.2	141.6	0.200616	0.260962	100%	100%
7	0.000274	0.000253	72	73.8	0.22966	0.197222	100%	100%
8	0.685057	0.291755	6096	569.4	0.054336	0.174963	100%	100%
9	0.000002	0.000001	60.6	64.6	0.179978	0.206901	100%	100%
10	0.001109	0.0011	60.6	60.6	0.161759	0.177124	100%	100%
11	0.60187	0.077945	11341.8	3139.8	0.067786	0.245377	100%	100%
12	0.022248	0.002793	78	87	0.243564	0.23989	100%	100%
13	0.001848	0.001648	1767	1767	0.253535	0.263535	100%	100%
14	0.000126	0.000109	60	60	0.048579	0.055405	100%	100%
15	0.000009	0.000003	60	60	0.005729	0.003796	100%	100%

Table 10- Comparison of minimum objective function value of SPSO and HPSO for 13 Non-Scalable Problem Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO	SPSO	HPSO
1	0.5	0.5	60	60	0.042453	0.042463	100%	100%
2	0.017193	0.002786	64.2	63	0.258362	0.299918	100%	100%
3	0.001029	0.001027	66.6	74.4	0.224219	0.236823	100%	100%
4	0.3986	0.390856	128.4	181.8	0.13771	0.148293	100%	100%
5	0.018613	0.01239	72	66	0.240972	0.232866	100%	100%
6	0.4986	0.4786	128.4	128.4	0.16771	0.13771	100%	100%
7	0.027193	0.012786	64.2	63	0.358362	0.309918	100%	100%
8	0.015341	0.014276	82.2	85.2	0.281294	0.257859	100%	100%
9	0.480507	0.48047	60	60	0.026709	0.023939	100%	100%
10	0.067997	0.037472	840.6	517.2	0.215576	0.251745	100%	100%
11	0.003378	0.003178	60.6	61.6	0.207517	0.227517	100%	100%
12	0.005549	0.00536	63.6	65.4	0.270722	0.275501	100%	100%
13	0.002655	0.002378	65.4	62	0.229666	0.21365	100%	100%

In observing [Table-10], it can be seen that HPSO gives a better quality of solutions as compared to SPSO. Thus, for the non-scalable problems HPSO outperforms SPSO with respect to efficiency, reliability, cost and robustness.

[Fig-3A] and [Fig-3B] shows Comparison of minimum objective function of SPSO and HPSO for 15 Scalable and 13 Non Scalable Problems Set-I and Set-II. [Fig-4A] and [Fig-4B] reflects Comparison of minimum objective function of SPSO and HPSO for 15 Scalable and 13 Non Scalable Problems Set-I and Set-II.[Fig-5A] and [Fig-5B] showing the comparison of minimum objective function of SPSO and HPSO for 15 Scalable and 13 Non Scalable Problems Set-I and Set-II.

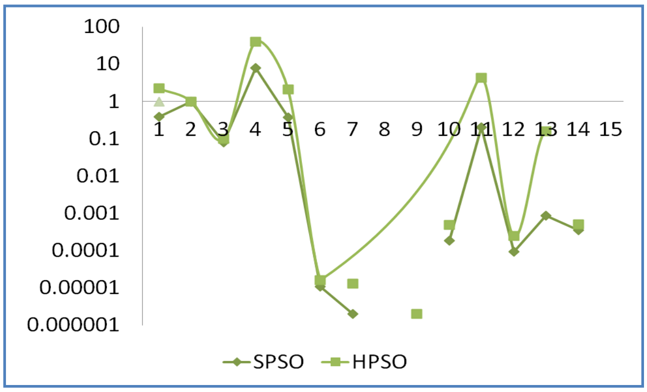

Fig. 3A-

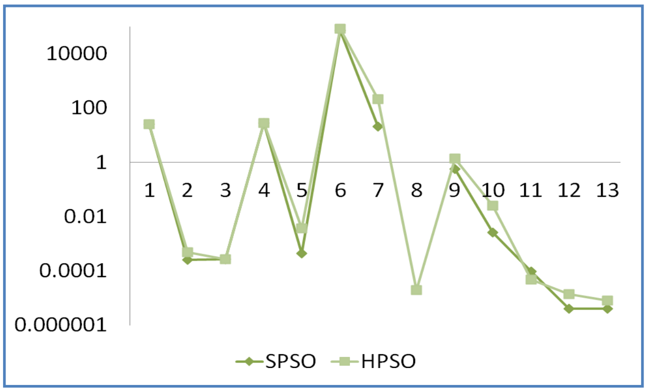

Fig. 3B-

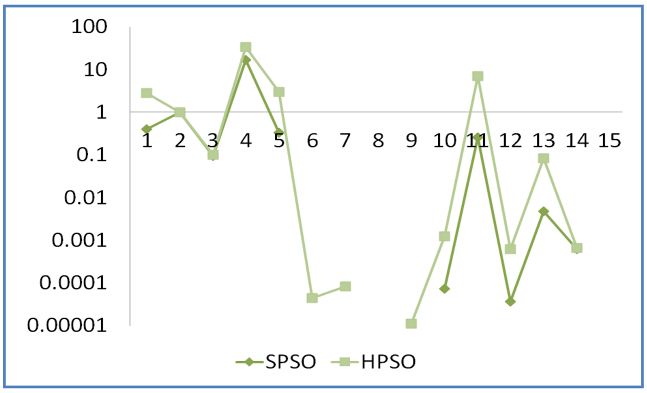

Fig. 4A-

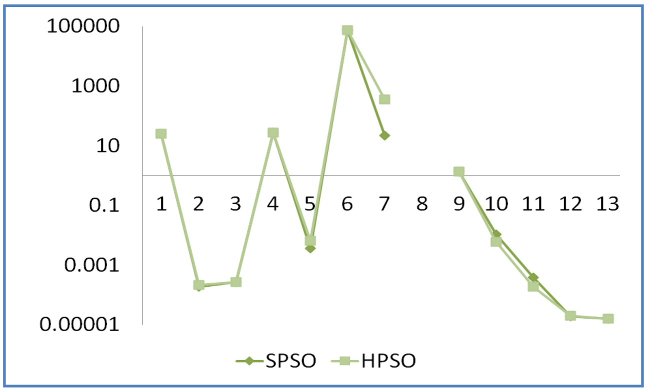

Fig. 4B-

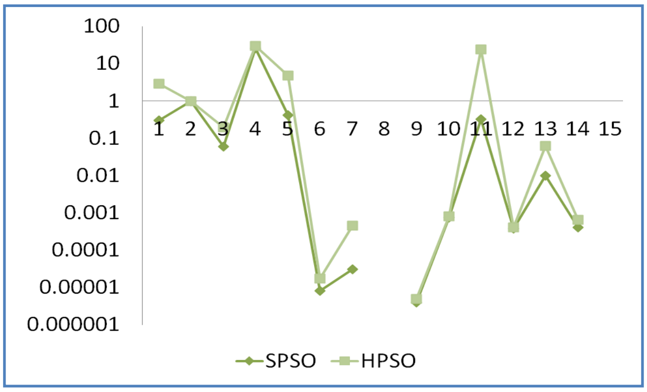

Fig. 5A-

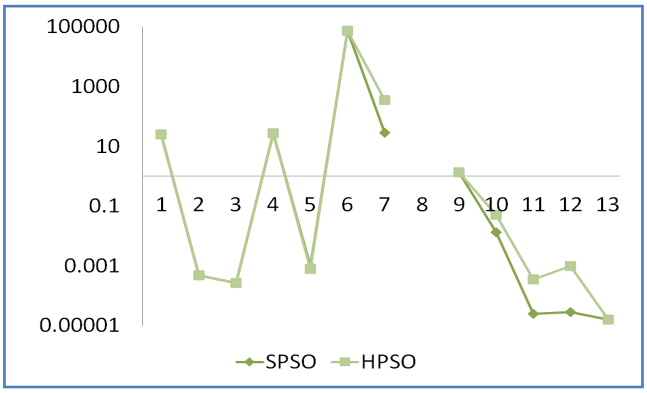

Fig. 5B-

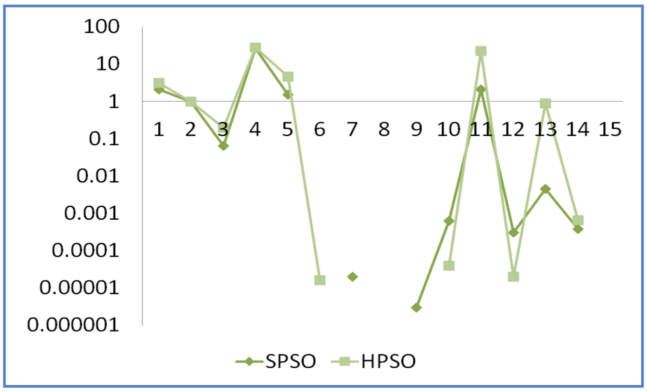

Fig. 6A-

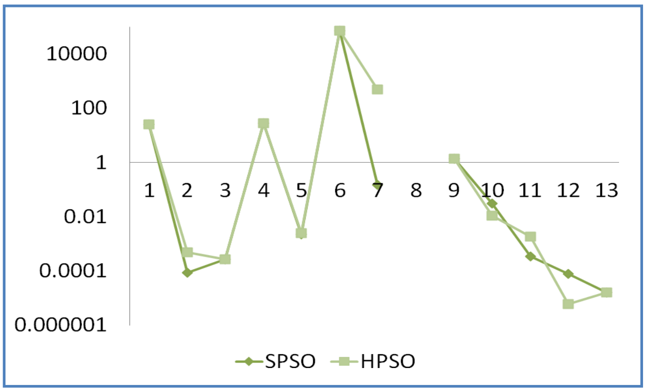

Fig. 6B-

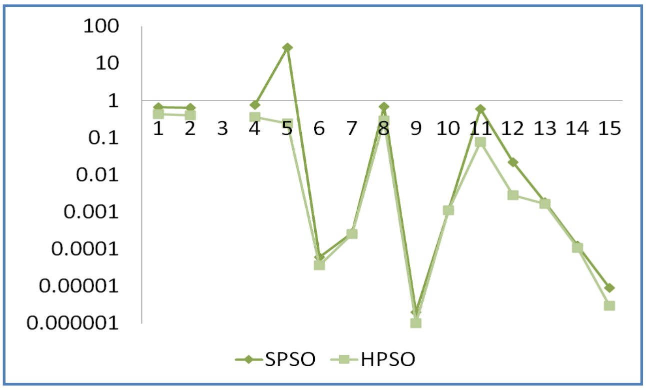

Fig. 7A-

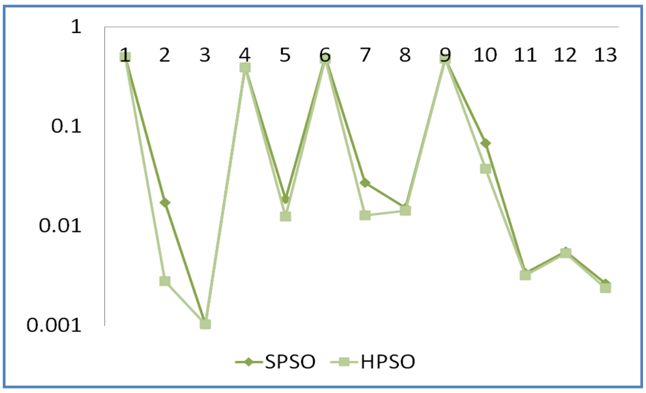

Fig. 7B-

[Fig-6A] and [Fig-6B] shows Comparison of minimum objective function of SPSO and HPSO for 15 Scalable and 13 Non Scalable Problems Set-I and Set-II. [Fig-7A] and [Fig-7B]Compares minimum objective function of SPSO and HPSO with the help of 15 Scalable and 13 Non-Problems SET-I and SET-II.

Conclusions

In this paper a hybrid Particle Swarm Optimization (HPSO) has been proposed by combining two different approaches of PSO *i.e.* Standard Particle Swarm Optimization and Mean Particle Swarm Optimization. The performance of HPSO algorithm has been tested on 28 benchmark Problems (15 Scalable and 13 Non-Scalable). Numerical experiments were performed to analyse the effect of inertia weight and acceleration coefficient on the performance of the algorithm. Proposed Algorithm with the values of parameters inertia weight 0.7 and acceleration coefficient 1.5 gives the best convergence. Other combination of parameters may in some cases lead to non- convergence. On the basis of results obtained it may be concluded that the newly proposed HPSO algorithm outperforms the classical SPSO algorithm in terms of convergence, speed and quality of the solution.

Nomenclature

C_1	Self Confidence Factor
C_2	Swarm Confidence Factor (The parameters C_1 and C_2 in equation (2), are not critical for PSO's convergence and alleviation of local minima, C_1 than a social parameter C_2 but with $C_1 + C_2 = 4$)
f	Fitness Function
y_{ij}	Personal Best Position of the i^{th} particle in j^{th} dimension
$\hat{y}_{ij}$	Global Best Position of the i^{th} particle in j^{th} dimension
$v_{ij}(t)$	Old Velocity of the i^{th} particle in j^{th} dimension
$v_{ij}(k+1)$	New Update Velocity i^{th} particle in j^{th} dimension
$X_{ij}(k)$	Old Position of the i^{th} particle in j^{th} dimension
$x_{ij}(k+1)$	New Update Position of the i^{th} Particles in j^{th} dimension
w	Inertia Weight:- (The role of inertia weight in equation (2), is considered critical for the PSO,s convergence behavior. The inertia weight is employed to control the impact of previous history of velocities on the current one.
χ	Constriction Coefficient
r	Random Number between 0 and 1(The parameters r_1 and r_2 are used to maintain the diversity of the population, and they are uniformly distributed in the range [0,1])
Vmax	Maximum velocity (Vmax) parameter. This parameter limits the maximum jump that a particle can make in one step.
R	Real Number
S	Swarm Size :-(Number of particles in the swarm affects the run-time significantly, thus a balance between variety (more particles) and speed (less particles) must be sought)
R^n	Real Number of n -triples
k	Time
Δk	Time Increment
$U(0,1)$	Uniformly distribution between 0 and 1
ϕ	Objective Function

Acknowledgements

Authors are very thankful to Dr. Chander Mohan, Professor of Computer Science, Ambala College of Engineering and Applied Research, Mithapur, Ambala (Former Professor of Mathematics at I.I.T. Roorkee) for his kind support and valuable suggestions.

References

[1] Eberhart R.C. and Kennedy J. (1995) *Sixth International Symposium on Micromachine and Human Science*, 39-43.

[2] Kennedy J. and Eberhart R.C. (1995) *IEEE International Joint Conference on Neural Networks, IEEE Press*, 1942-1948.

[3] Chatterjee A., Pulasinghe K., Watanabe K., et al. (2005) *IEEE Trans. on Industrial Electronics*, 52, 1478-1489.

[4] Kennedy J. and Mendes R. (2002) *IEEE Congress on Evolutionary Computation, IEEE Press*, 1671-1676.

[5] Peer E.S., Van den Bergh F. and Engelbrecht A.P. (2003) *IEEE Swarm Intelligence Symposium, IEEE Press*, 235-242.

[6] Engelbrecht A.P. (2005) *Fundamentals of Computational Swarm Intelligence, Wiley & Sons*.

[7] Kennedy J., Eberhart R.C., and Shi Y. (2001) *Swarm Intelligence, Morgan Kaufmann*.

[8] Van den Bergh F. (2002) *An Analysis of Particle Swarm Optimizers, Ph.D. thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa*.

[9] Van den Bergh F. and Engelbrecht A.P. (2006) *Information Sciences*, 176(8), 937-971.

[10] Kennedy J. (2003) *IEEE Swarm Intelligence Symposium*, 80-87.

[11] Shi Y. and Eberhart R.C. (1998) *IEEE Congress on Evolutionary Computation*, 69-73.

[12] Angline P.J. (1998) *Lecture Notes in Computer Science*, 1447, 601-610.

[13] Angline P.J (1998) *IEEE Conference on Evolutionary Computations*, 84-89.

[14] Banks A., Vincent J. and Anyakoha C. (2007) *Natural Computing: an International Journal*, 6(4), 467-484.

[15] Banks A., Vincent J. and Anyakoha C. (2008) *Natural Computing: an International Journal*, 7(1), 109-124.

[16] Ziyu T. and Dingxue Z. (2009) *Asia-Pacific Conference on Information Processing*.

[17] Singh Narinder and Singh S.B. (2011) *International Journal of Scientific and Engineering Research*, 2(8).

[18] Zhan Z.H., Zhang J., Li Y. and Chung H.S.H (2009) *IEEE Transactions on Systems, Man and Cybernetics*, 1362-1381.

[19] Xinchao Z. (2010) *Applied Soft Computing*, 119-124.

[20] Niknam T. and Amiri B. (2010) *Applied Soft Computing*, 183-197.

[21] Abda M.El., Hassan H., Anisa M., Kamela M.S., and Elmasry M. (2010) *Applied Soft Computing*, 284-295.

[22] Chena M.R., Lia X., Zhanga X., and Lu Y.Z. (2010) *Applied Soft Computing*, 367-373.

[23] Tsang P.W.M., Yuena T.Y.F., and Situ. W.C. (2010) *Applied Soft Computing*, 432-438.

[24] Hsua C.C., Shiehb W.Y., and Gao C.H. (2010) *Applied Soft Computing*, 606-612.

[25]Liua H., Caia Z. and Wang Y. (2010) *Applied Soft Computing*, 629-640.

[26]Mahadevana K. and Kannan P.S. (2010) *Applied Soft Computing*, 641-652.

[27]Pedersen M.E.H. (2010) *Tuning & Simplifying Heuristical Optimization," Ph.D. thesis, School of Engineering Sciences, University of Southampton, England*.

[28]Pedersen M.E.H. and Chipper A.J. (2010) *Applied Soft Computing*, 618-628.

[29]Montes de Oca M.A. (2007) *Institut de Recherches Interdisciplinairrs et de Developpements en Intelligence Artificielle*.

[30]Schutte J.F. and Groenwold A.A. (2005) *Journal of Global Optimization*, 31(1), 93-108.

[31]Pedersen M.E.H. and Chipper A.J. (2010) *Applied Soft Computing*, 10(2), 618-628.

[32]Sedighizadeh D. and Masehian E. (2009) *International Journal of Computer Theory and Engineering*, 1(5), 1793-8201.

[33]Clerc M. and Kennedy J. (2002) *IEEE Trans. Evol. Computer*, 6 (1), 58-73.

[34]Eberhart R.C. and Shi Y. (2000) *IEEE Congress Evolutionary Computation, San Diego*, *CA*, 84-88.

[35]Deep K. and Bansal J.C. (2009) *Int. J. Computational Intelligence Studies*, 1(1).