International Journal Of Scientific & Engineering Research Volume 3, Issue 8, August-2012 1
ISSN 2229-5518

Half Mean Particle Swarm Optimization Algorithm

Narinder Singh, Sharandeep Singh, S.B. Singh and Shelly Arora

Abstract -This paper introduces a Half Mean Particle Swarm Optimization algorithm (HMPSO) and discusses the results of experimentally comparing the performances of SPSO. This is done by replacing one term of original velocity update equation by one 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 29 benchmark test problems (15 Scalable and 13 Non-Scalable Problems). Based on the numerical and graphical analyses of results it is shown that the HMPSO outperforms the SPSO (Standard Particle Swarm Optimization), in terms of efficiency, reliability, accuracy and stability.

Index Terms: SPSO, HMPSO, global optimization, velocity update equation, pbest (personal best position), gbest (global best position).

1. INTRODUCTION

THE particle swarm algorithm, which is frequently called particle swarm optimizer, is a new evolutionary algorithm, where the population is called a swarm and each individual is called a particle [1]. It is inspired by the behavior of bird flocking and fish schooling. A large number of birds or fish flock synchronously, change direction suddenly, and scatter and regroup together.

***Prof. (Dr.) S.B. Singh**, Department of Mathematics, Punjabi University, Patiala, INDIA, Punjab, Pin No- 147002, Email ID: sbsingh69@yahoo.com. Presently the author is heading the Department. The research interests of the author are Mathematical Modeling and Optimization Techniques. He has more than 50 research papers in national and International Journals. He has received Khosla Gold Medal for outstanding research contributions.*

Dr. Shelly Arora, Department of Mathematics, Punjabi University, Patiala, Pin Code-147002,Email ID: shellyarora_25@yahoo.co.in. The auther is Assistant Professor in the Department of Mathematics, PUP. The research interests of author are Mathematical Modeling and Numerical Analysis. She has more than 21 research papers in journals/ Conferences.

***Mr. Narinder Singh**, Department of Mathematics, Punjabi University, Patiala, INDIA, Punjab, Pin No- 147002, Email ID: narindersinghgoria@yahoo.com. He is currently a Ph.D. student at the Department of Mathematics, Punjabi University, India, Punjab, Patiala-147002. His area of research interests is in Particle Swarm Optimization.*

***Sh. Sharandeep Singh**, Department of Mathematics, Punjabi University, Patiala, INDIA, Punjab, Pin No- 147002, Email ID: sssinghsharan@gmail.com. He is currently a Ph.D. student at the Department of Mathematics, Punjabi University, India, Punjab, Patiala-147002. His area of research interests is in Particle Swarm Optimization.*

Each particle benefits from the experience of its own and that of the other members of the swarm during the search for food.

Particle Swarm Optimization (PSO) algorithm was proposed by Eberhart engineering [1]. Since its inception the algorithm has been applied in various disciplines of science and engineering [2, 3, 4, 5, 6].

The convergence and parameterization aspects of the PSO have also been discussed by various authors [7, 8, 9]. Particle swarm optimization [1, 2] is a stochastic, population-based search method, modeled after the behavior of bird flocks. 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 are adjusted using

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

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

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

where w is the inertia weight [11], c_1 *and* c_2 are the acceleration coefficients, $r_{1j}, r_{2j} \sim U(0,1)$, y_{ij} is the personal best position of particle i, and $\hat{y}_j$ is the neighborhood best position of particle i.

One Half Global Best Position Particle Swarm Optimization is introduced by Narinder Singh and S. B. Singh [10]. The performance of this algorithm has been tested through numerical and graphical results. The results obtained are compared with the standard PSO (SPSO) for scalable and non-scalable problems.

Personal Best Position Particle Swarm Optimization is introduced by Narinder Singh and S.B. Singh [11]. In this

ISSN 2229-5518

proposed approach a novel philosophy of modifying the velocity update equation of Standard Particle Swarm Optimization approach has been used. The modification has been done by vanishing the gbest term in the velocity update equation of SPSO. The performance of this proposed algorithm (Personal Best Position Particle Swarm Optimization, PBPPSO) has been tested on several benchmark problems. It is concluded that the PBPPSO performs better than SPSO in terms of accuracy and quality of solution.

A New Version of Particle Swarm Optimization Algorithm developed by Narinder Singh and S.B.Singh [12]. In this paper an algorithm has been developed by combining two different approaches of PSO i.e., Standard Particle Swarm Optimization and Mean Particle Swarm Optimization. Numerical experiments for scalable and non-scalable well known test problems have shown the superiority of newly proposed Hybrid Particle Swarm Optimization (HPSO) approach, compared with the classical SPSO algorithm in term of convergence, speed and quality of obtained solutions.

Jun-qing Li et al. [13] proposed a hybrid algorithm with particle swarm optimization (PSO) and tabu search algorithm (TS) for solving the FJSP problems. Some novel approaches are introduced in the hybrid algorithm: a new chromosome representation for the FJSP solutions is presented; some novel crossover and mutation functions for the generation evolutionary are developed. In each generation, they used tabu search algorithm to find near optimum solutions for the obtained best solution.

Davoud Sedighizadeh et al. [14] carried out an overview of previous and present conditions of the PSO algorithm as well as its opportunities and challenges. Accordingly, the history, various methods, and taxonomy of this algorithm are discussed and its different applications together with an analysis of these applications are evaluated.

Jun-qing Li et al. [15] had given a new chromosome representation for the FJSP solutions, and proposed some novel crossover and mutation functions for the particle swarm optimization algorithm. In each generation, they used tabu search algorithm to find near optimum solutions for the given best solution. After a detailed experiment, verification of the results, they stated that their novel method can get better solutions in very short period.

J. Zhang et al. [16] studied the impact of Particle Swarm Optimization on the evaluation of cooperation in the stochastic strategy spatial prisoner's dilemma game. The strategy updating was guided by the particle swarm optimization algorithm, using as input the individual memory of every player as well as knowledge gained by the swarm as a whole.

S. J. Bassi et al. [17] presented an artificial intelligence (AI) method of particle swarm optimization (PSO) algorithm for tuning the optimal proportional-integral derivative (PID) controller parameters for industrial processes. This approach has superior features, including easy implementation, stable convergence characteristic and good computational efficiency over the conventional methods.

Taoshen LI et al. [18] proposed a new communication mode in IPv6. The simulation experiments show that the algorithm is feasible and effective in any cast routing. It can effectively break away from the local optimum and improve the convergence velocity extraordinarily.

Pinar Civicioglu et al. [19] proposed the concept of the Cuckoo-search (CK), Particle swarm optimization (PSO), Differential evolution (DE) and Artificial bee colony (ABC) algorithms. The performances of the CK and PSO algorithms are statistically closer to the performance of the DE algorithm than the ABC algorithm. The CK and DE algorithms supplymore robust and precise results than the PSO and ABC algorithms.

2. THE NEW PROPOSED HALF MEAN PSO

The motivation behind introducing HMPSO is that in the velocity update equation instead of comparing the particle's current position with gbest and pbest, it is compared with linear combination y_{ij} and $\frac{(y_{ij} - \hat{y}_j)}{2}$ of pbest and gbest. Thus, we introduce a new velocity update equation as fellows:

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

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 $(y_{ij} - x_{ij})$ and is responsible for the attractor of particle's current position. The third term is proportional to the vector $\left(\frac{(y_{ij} - \hat{y}_j)}{2} - x_{ij}\right)$, is responsible for the attractor of particle's current position towards the mean of the positive direction of its own best position (pbest) and the negative direction of the global best position (-gbest). Clearly, HMPSO seems to be suitable name for this modified PSO. The relative position of the position generated by SPSO and HMPSO can be visualized in Figure I.

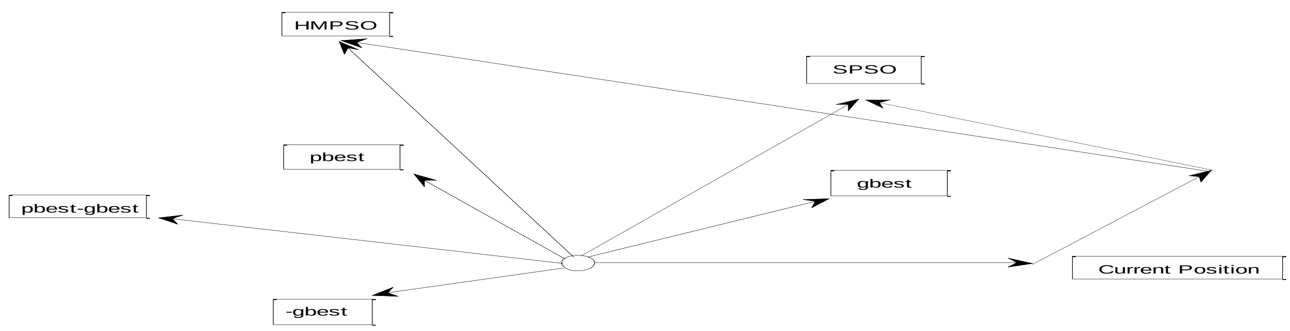

Figure I: Movement of SPSO and HPSO

Note: - - - - Movement of SPSO & --------- Movement of HMPSO.

The pseudo code of HMPSO is shown below:

ALGORITHM- HMPSO

I) For each particle:

Initialize particle

II) Do:

a) For each particle:

- Calculate fitness value
- If the fitness value is better than the best fitness value (pbest) in history
- Set current value as the new pbest End

b) For each particle:

- Find in the particle neighborhood, the particle with the best fitness
- Calculate particle velocity according to the velocity equation (5)
- Update particle position according to the position equation (1)

END ALGORITHM

3. TEST PROBLEMS

Many times it is found that the evaluation of a proposed algorithm is evaluated only on a few benchmark problems. However, in this paper we consider a test bed of thirty benchmark problems with varying difficulty levels and problem size. The relative performance of SPSO and HMPSO is evaluated on two kinds of problem sets. Problem Set 1 consists of 15 scalable problems, i.e., those problems in which the dimension of the problems can be increased / decreased at will.

In general, the complexity of the problem increases as the problem size is increased. Problem Set 2 consists of those problems in which the problem size is fixed, but the problems have many local as well as global optima. The Problem Set 1 is shown in Table 1 and Problem Set 2 is shown in Table 2.

Table-1: Detail of 15 Scalable Problems SET-I (Continued)
(In which Particle size in the swarm increasing and decreasing, no particle sized is fixed).

Problem No.	Problems Name	Problems	Range of the Problems
1.	Ackley	$Min\ f(x) = -20\exp(-0.02\sqrt{n^{-1}\sum_{i=1}^{n} x_i^2}) - \exp(n^{-1}\sum_{i=1}^{n}\cos(\pi x_i)) + 20 + e$	In which search space lies between $-30 \le x_i \le 30$ and Min Objective Function Value is 0.

2.	**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 Min Objective Function Value is $-0.1\times(n)$.
3.	**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 Min Objective Function Value is -1.
4.	**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 Min Objective Function Value is 0.
5.	**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 Min Objective Function Value is 0.
6.	**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 Min Objective Function Value is 0.
7.	**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 Min Objective Function Value is 0.
8.	**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 Min Objective Function Value is 0.
9.	**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 Min Objective Function Value is 0.
10.	**Schwefel '3'**	$Min\ f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	In which search space lies between $-10 \le x_i \le 10$ and Min Objective Function Value is 0.
11.	**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 Min Objective Function Value is 0.
12.	**Schwefel '4'**	$Min\ f(x) = Max\{\lvert x_i\rvert, 1 \le i \le n\}$	In which search space lies between $-100 \le x_i \le 100$ and Min Objective Function Value is 0.
13.	**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 Min Objective Function Value is 0.
14.	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 Min Objective Function Value is 0.

15.	**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 Min Objective Function Value is 0.

Table-2: Detail of 13 Non- Scalable Problems SET-II

((In which Particle size in the swarm is fixed, no particle increasing and decreasing in the swarm).

Problem No.	Problems Name	Problems	Range
1.	**Becker and Lago**	$Min\ f(x) = (\lvert x_1 \rvert - 5)^2 + (\lvert x_2 \rvert - 5)^2$	In which search space lies between $-10 \le x_i \le 10$ and Min Objective Function Value is 0.
2.	**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 Min Objective Function Value is 0.
3.	**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 Min Objective Function Value is 0.
4.	**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 Min Objective Function Value is 0.398.
5.	**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 Min Objective Function Value is 0.
6.	**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 Min Objective Function Value is 0.
7.	**Modified Rosenbrock**	$Min\ f(x) = 100(x_2 - x_1^2)^2 + [6.4(x_2 - 0.5)$	In which search space lies between $-5 \le x_1, x_2 \le 5$ and Min Objective Function Value is 0
8.	**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 Min Objective Function Value is -1

9.	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 Min Objective Function Value is 0.9
10.	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 Min Objective Function Value is 0
11.	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 Min Objective Function Value is 0
12.	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 Min Objective Function Value is -1.0316
13.	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 Min Objective Function Value is 0.352

4. ANALYSES OF RESULTS AND DISCUSSION

The SPSO and the HMPSO are coded in C++ and implemented on Pentium-IV 2.4 GHz machine with 512 MB RAM under WINXP platform. Thirty independent runs with different seed for the generation of random numbers are taken. However, the same seed is used for generating the initial swarm for SPSO and HMPSO for the i^{th} run, where $i = 1, 2, ..., 50$. A run is said to be a successful run if the best objective function value found in that run lies within 1% accuracy of the best known objective function value of the problem. The maximum number of function evaluations is fixed to be 30,000. The swarm size is fixed to 20 and dim is 30. The inertia weight is 0.7 and the acceleration coefficients for SPSO and HMPSO are set to be $c_1 = c_2 = 1.4$.

A number of criterions are used to evaluate the performance of SPSO and HMPSO. 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-I, by fixing for problem measured by the minimum, mean, success of rate and standard deviation of the objective function values out of fifty runs. This is shown in Table 3. The corresponding information for problem SET-II is shown Table 4.

This new approach testing on different type of parameter setting. Firstly, We are setting the parameter maximum number of function evaluations is fixed to be 30,000, swarm size is 20 and dim is 30, inertia weight is 0.6 and the acceleration coefficients for SPSO and HMPSO are set to be $c_1 = c_2 = 1.3$. With the help of this parameter setting we have found the optimal results and compared them with both the techniques SPSO and HMPSO. But this comparison of results show that both approach (SPSO and HMPSO) have been failed to find the global optimal point.

Secondly, We are setting the parameter maximum number of function evaluations is fixed to be 30,000, swarm size is 20 and dim is 30, inertia weight is 0.7 and the acceleration coefficients for SPSO and HMPSO are set to be $c_1 = c_2 = 1.4$. With the help of this parameter setting we have found the optimal results and compare the results of both the technique SPSO and HMPSO. According to this parameter setting the new approach gives the better optimal results as comparison to Standard Particle Swarm Optimization Algorithm.

ISSN 2229-5518

In observing Table 3, it can be seen that HMPSO gives a better quality of solutions as compared to SPSO. Thus, for the scalable problems HMPSO outperforms SPSO in terms of efficiency, reliability, cost and robustness.

In observing Table 4, it can be seen that HMPSO gives a better quality of solutions as compared to SPSO. Thus, for the non-scalable problems HMPSO outperforms SPSO with respect to efficiency, reliability, cost and robustness.

In Table 3, It is observed that SPSO could not solve two problems with 100% success, whereas HPSO solved all the problems with 100% success.

For justification of the proposed algorithm the approach has been tested with the following parameters also. The maximum number of function evaluations is fixed to be 30,000, swarm size is 20 and dim is 30, inertia weight is 0.8, 0.9 and the acceleration coefficients for SPSO and HMPSO are set to be $c_1 = c_2 = 1.5, 1.6$) on this parameter setting we are finding the global optimal results. These optimal results shown that both techniques SPSO and HMPSO are failed on different types of problems (i.e. Scalable and Non- Scalable Problems) by setting of this parameter setting.

The results show that the proposed algorithm HMPSO and SPSO are failed for several scalable as well as non-scalable problems. Thus authors conclude that the parameter setting i.e. maximum number of function evaluations is fixed to be 30,000, swarm size is 20 and dim is 30, inertia weight is 0.7 and the acceleration coefficients for SPSO and HMPSO are set to be $c_1 = c_2 = 1.4$ is most appropriate.

Table-3 Comparative Objective function value obtained in 50 runs by SPSO and HMPSO for problem Set-I

Problem No.	**Minimum Function Value**		**Mean Function Value**		**Standard Deviation**		**Rate of Success**	
	SPSO	HMPSO	SPSO	HMPSO	SPSO	HMPSO	SPSO	**HMPSO**
1	0.838771	0.243638	25306.80000	917.400000	0.829384	0.157145	**18.00%**	**100%**
2	0.667680	0.507389	468.600000	201.600000	0.067679	0.115498	100%	**100%**
3	0.000000	0.000000	60.000000	60.000000	0.000213	0.000247	100%	**100%**
4	0.706757	0.402129	4204.800000	1041.000000	0.032048	0.117709	100%	**100%**
5	19.899176	0.306378	30000.000000	1174.200000	12.279927	0.156808	**0.00%**	**100%**
6	0.000278	0.000212	109.800000	98.200000	0.209093	0.233719	100%	**100%**
7	0.000046	0.000033	66.600000	76.200000	0.270036	0.263288	100%	**100%**
8	0.693786	0.286580	1697.400000	495.600000	0.056064	0.157950	100%	**100%**
9	0.000006	0.000004	60.600000	61.800000	0.136879	0.186732	100%	**100%**
10	0.008521	0.001894	60.600000	60.600000	0.169329	0.189341	100%	**100%**
11	0.648966	0.033271	3223.200000	675.000000	0.059425	0.227979	100%	**100%**
12	0.008174	0.007901	73.800000	84.000000	0.269678	0.265533	100%	**100%**
13	0.003685	0.001921	844.800000	954.800000	0.225323	0.273223	100%	**100%**
14	0.000126	0.000124	60.000000	60.000000	0.043944	0.075360	100%	**100%**
15	**0.000004**	**0.000000**	**60.000000**	**60.000000**	**0.007078**	**0.004230**	**100%**	100%

Table-4 Comparative Objective function value obtained in 50 runs by SPSO and HMPSO for problem Set-II

Problem No.	**Minimum Function Value**		**Mean Function Value**		**Standard Deviation**		**Success of Rate**	
	SPSO	HMPSO	SPSO	HMPSO	SPSO	HMPSO	SPSO	**HMPSO**
1	0.500000	0.500000	60.000000	61.800000	0.035763	0.075067	100%	**100%**
2	0.017608	0.006437	62.400000	70.800000	0.214381	0.240656	100%	**100%**
3	0.003310	0.002117	67.200000	70.200000	0.244361	0.267334	100%	**100%**
4	0.012269	0.003965	87.600000	109.200000	0.282993	0.248732	100%	**100%**
5	0.024190	0.001515	70.200000	73.800000	0.216831	0.246397	100%	**100%**
6	0.001200	0.000117	90.200000	97800000	0.295838	0.286377	100%	**100%**
7	0.000319	0.000117	67.200000	70.200000	0.204370	0.267395	100%	**100%**

8	0.008103	0.006402	72.000000	88.200000	0.238262	0.238889	100%	**100%**
9	0.480466	0.480455	60.000000	60.000000	0.023049	0.030635	100%	**100%**
10	0.003745	0.001902	436.200000	439.800000	0.267266	0.296418	100%	**100%**
11	0.007222	0.005314	61.800000	64.800000	0.189268	0.190946	100%	**100%**
12	0.028283	0.006380	66.000000	72.000000	0.228904	0.239190	100%	**100%**
13	**0.005644**	**0.004994**	**61.200000**	**63.000000**	**0.186620**	**0.210144**	**100%**	**100%**

Figure A : Comparison of SPSO and HMPSO with the help of Scalable 15 Problems SET-I.

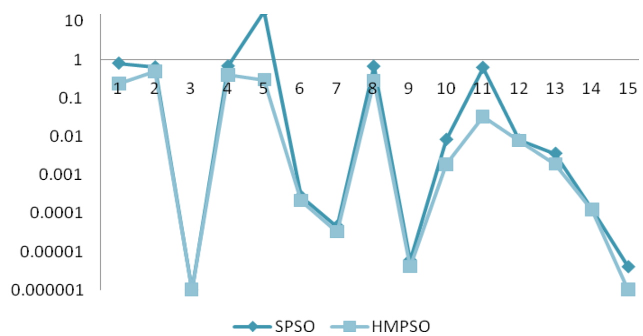

Figure B : Comparison of SPSO and HMPSO with the help of Non-Scalable 13 Problems SET-II.

5. CONCLUSIONS

On the basis of the numerical results authors conclude that the proposed algorithm HMPSO outperforms SPSO in terms of efficiency, reliability, cost and robustness.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", *Proceeding 6th Int. Symp. Micro Machine and Human Science, Nagoya, Japan*, pp. 39–43, 1995.

[2] Y. Shi and R. Eberhart, "A Combined Particle Swarm Optimizer", *Proceedings of IEEE World Congress on Computational Intelligence*, pp. 69–73, 1998.

[3] V. Tandon, "Closing The Gap Between CAD/CAM and Optimized CNC and Milling", *Master thesis, Purdue School of Engineering and Technology*, Indiana University Purdue University Indianapolis, 2000.

[4] Abido M.A., "Optimal Power Flow Using Particle Swarm Optimization", Electri Power and Energy System, vol. 24, pp. 563–71, 2002.

[5] Jiang Chuanwen, Etorre Bompard, "A Hybrid Method Of Chaotic Particle Swarm Optimization and Linear Interior For Reactive Power Optimization", *Mathematics and Computers in Simulation* , vol. 68 , pp. 57–65, 2005.

[6] N. Shigenori, G. Takamu, Y. Toshiku, F. Yoshikazu, "A Hybrid Particle Swarm Optimization For Distribution State Estimation", *IEEE Transactions on Power Systems*, vol. 18 , pp. 60– 68, 2003.

[7] R. C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors In Particle Swarm Optimization" *In Proceeding CEC, San Diego*, CA, pp. 84–88, 2000.

[8] Ioan Cristian Trelea, "The Particle Swarm Optimization Algorithm: Convergence Analysis And Parameter Selection", *Information Processing Letters*, vol. 85 , pp. 317–325, 2003.

[9] R. Eberhart and Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", *The 7th Annual Conference on Evolutionary Programming*, San Diego, USA, 1998.

[10] Narinder Singh and S.B.Singh, " One Half Global Best Position Particle Swarm Optimization", *International Journal of Scientific & Engineering Research*, vol. 2, no. 8,: ISSN 2229-5518, 2011 .

[11] Narinder Singh and S.B.Singh,"Personal Best Position Particle Swarm Optimization", *Journal of Applied Computer Science & Mathematics*, Suceava, vol. 12, no.6, 2012,

[12] Narinder Singh and S.B.Singh, "A New Version of Particle Swarm Optimization Algorithm", *Accepted in Journal of Artificial Intelligence, bio informatics, EBSCO Publishing*, USA; Ovid Technologies, USA; ProQuest, USA, 2012

[13] Jun-qing Li et al, "An effective hybrid particle swarm optimization algorithm for flexible job-shop scheduling problem", *MASAUM Journal of Computing* , vol.1 no.1, pp. 69-74, 2009.

[14] Jun-qing Li, Quan-ke Pan, Sheng-xian Xie, Bao-xian Jia & Yu-ting Wang, " A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem ",*International Journal of Computer Theory and Engineering*, vol. 2, no. 2, pp.1793-1801, 2010.

[15] Davoud Sedighizadeh and Ellips Masehian, "Particle Swarm Optimization Methods, Taxonomy and Applications", *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, pp. 1793-8201, 2009.

[16] J. Zhang, C. Zhang, T. Chu, and M. Perc, "Resolution of the Stochastic Strategy Spatial Prisoner's Dilemma by Means of Particle Swarm Optimization", *PloS ONE, www.plosone.org*, vol 6, no. 7, 2011.

[17] S. J. Bassi, M. K. Mishra & E. E. Omizegba, "Automatic Tuning Of Proportional Integral Derivative (PID) Controller Using Particle Swarm Optimization (PSO) Algorithm", *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol.2, no.4, October 2011.

[18] Taoshen LI, Zhigang ZHAO, Zhihui GE, "An Adaptive Particle Swarm Optimization Algorithm for Anycast Routing" *Journal of Computational Information Systems, Available at http://www.Jofcis.com*, vol. 7, no. 5 , pp.1559-1566, 2011.

[19] Pinar Civicioglu & Erkan Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms", *Artificial Intelligence Rev DOI 10.1007/s 10462-011-9276-0*, 2011.