One Half Global Best Position Particle Swarm Optimization Algorithm

Narinder Singh, S.B. Singh

Abstract-In this paper, a new particle swarm optimization algorithm have been proposed. The algorithm is named as One Half Personal Best Position Particle Swarm Optimizations (OHGBPPSO) and a novel philosophy by modifying the velocity update equation has been presented. The performance of 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.

Index Terms- Particle Swarm Optimization, One Half Global Best Position Particle Swarm Optimization, Personal Best Position, Global Best Position, Global optimization, Velocity update equation.

1 INTRODUCTION

STANDARD Particle Swarm Optimization: Particle swarm optimization (PSO) [1] 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

$$\underbrace{x_i(t+1)}_{\text{Update Position}} = \underbrace{x_i(t)}_{\text{Previous Position}} + \underbrace{v_i(t+1)\Delta t}_{\text{New Update Velocity}} \quad ...(1)$$

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

$$\underbrace{v_{ij}(t+1)}_{\text{Update Velocity}} = \underbrace{wv_{ij}(t)}_{\text{Current Motion}} + \underbrace{c_1 r_{1j}\frac{(y_{ij}-x_{ij})}{\Delta t}}_{\text{Conitive Component}} + \underbrace{c_2 r_{2j}\frac{(\hat{y}_j-x_{ij})}{\Delta t}}_{\text{Social Compoent}} \quad ...(2)$$

where w is the inertia weight [12], c_1 *and* c_2 are the acceleration coefficients (first acceleration coefficient is represent the how much confidence in itself and second acceleration coefficient is referred the how much confidence in its neighborhood), $r_{1j}, r_{2j} \in U(0,1)$, y_{ij} is the personal best position of particle i and dimension j, and $\hat{y}_j$ is the neighborhood best position of particle i and dimension j. The Δt is represent the rate of change in time.

Eberhart and Shi [15] suggested a more generalized PSO, where a constriction coefficient is applied to both terms of the velocity formula. Clerc and Kennedy [14] showed that the constriction PSO can converge without using **Vmax**:

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

where the constriction factor was set 0.7289. By using the constriction coefficient, the amplitude of the particle's oscillation decreases, resulting in its convergence over time. Kennedy [10] carried out some experiments using a PSO variant, which drops the velocity term from the PSO equation.

If pi and pg were kept constant, a canonical PSO samples the search space following a bell shaped distribution centered exactly between the pi and pg.

This bare bones PSO produces normally distributed random numbers around the mean $\frac{p_{id} \oplus p_{gd}}{2}$ (for each dimension *d*), with the standard deviation of the Gaussian distribution being $|p_{id} \oplus p_{gd}|$.

Mendes and Kennedy [4] found that von Neumann topology (north, south, east and west, of each particle

Dr S.B. Singh, is Professor and Head in Department of Mathematics, Punjabi University, Patiala, INDIA, Punjab, Pin No- 147002, Email ID:sbsingh69@yahoo.com. The research interests of author are Mathematical Modeling and Optimization Techniques. He has published 50 research papers in Journals/Conference Proceedings. He was awarded Khosla Gold Medal by I.I.T. Roorkee for his outstanding research contributions.
Mr. Narinder Singh is Ph.D. student in Department of Mathematics, Punjabi University, Patiala, Punjab, INDIA, Pin No- 147002, Email ID: narindersinghgoria@ymail.com.

placed on a 2 dimensional lattice) seems to be an overall winner among many different communication topologies.

Kennedy [10] also proposed an alternative version of the barebones PSO, where

$$v_{ij}(t+1)=\begin{cases} y_{ij} & \text{if } U(0,1) \prec 0.5 \\ N\left(\frac{y_{ij}+\hat{y}_{ij}}{2},\sigma\right) & \text{otherwise} \end{cases} \quad ...(3)$$

Based on equation (3), there is a 50% chance that the jth dimension of the particle dimension changes to the corresponding personal best position. This version of the barebones PSO biases towards exploiting personal best positions.

PSO variants are continually being devised in an attempt to overcome this deficiency, see e.g. [16] [17] [18] [19] [20] [21] [22] [23] [24] for a few recent additions. These PSO variants greatly increase the complexity of the original method and Pedersen and co workers [25, 26] have demonstrated that satisfactory performance can be achieved with the basic PSO if only its parameters are properly tuned.

2 THE NEW PROPOSED ALGORITHM

The motivation behind introducing OHGBPPSO is that in the velocity update equation instead of modifying the Personal Best and Global Best Position. We introduce a new velocity update equation as follows:

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}\frac{((y_{ij}+\frac{\hat{y}_j}{2})-x_{ij})}{\Delta t}+c_2r_{2j}\frac{((y_{ij}-\frac{\hat{y}_j}{2})-x_{ij})}{\Delta t}$$

$$OR$$

$$v_{ij}(t+1)= wv_{ij}(t)+c_1r_{1j}\frac{(\frac{2y_{ij}+\hat{y}_j}{2}-x_{ij})}{\Delta t}+c_2r_{2j}\frac{(\frac{2y_{ij}-\hat{y}_j}{2}-x_{ij})}{\Delta t} \quad ...(4)$$

where w is the inertia weight, c_1 and c_2 are the acceleration coefficients (first acceleration coefficient represent the how much confidence in itself and second acceleration coefficient referred the how much confidence in its neighborhood), $r_{1j}, r_{2j} \in U(0,1)$, y_{ij} is the personal best position of i particle and j dimension, and $\hat{y}_j$ is the neighborhood best position of particle i and dimension j. The Δt is represent the rate of change in time.

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 $\frac{c_1r_{1j}(\frac{(2y_{ij}+\hat{y}_j)}{2}-x_{ij})}{\Delta t}$, 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 $\frac{c_2r_{2j}(\frac{(2y_{ij}-\hat{y}_j)}{2}-x_{ij})}{\Delta t}$, is responsible for the attractor of particle's current position.

The algorithm of OHGBPPSO is shown below:

ALGORITHM- OHGBPPSO

- Randomly initialize particle position and Velocities.

- While do not terminate

  - Evaluate fitness objective functional value at current position x_i.
  - If objective functional value is better than Personal Best Position (y_{ij}) then update y_{ij}.
  - If objective function value is better than Global Best Position ($\hat{y}_j$) then update $\hat{y}_j$.

- For each particle;
  - Update velocity $v_{ij}(t+1)$ and position $x_{ij}(t+1)$

$$v_{ij}(t+1)=wv_{ij}(t)+\underbrace{\frac{c_1r_{1j}((y_{ij}+\frac{\hat{y}_j}{2})-x_{ij})}{\Delta t}}+\underbrace{\frac{c_2r_{2j}((y_{ij}-\frac{\hat{y}_j}{2})-x_{ij})}{\Delta t}}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1)$$

END OF THE ALGORITHM

Figure-I: Comparison of Particle Movement of SPSO and OHGBPPSO by Scalable Problems.

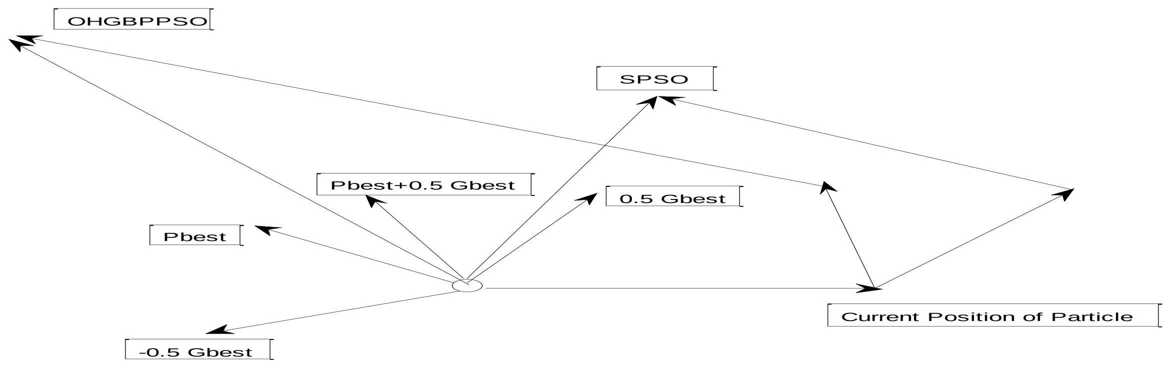
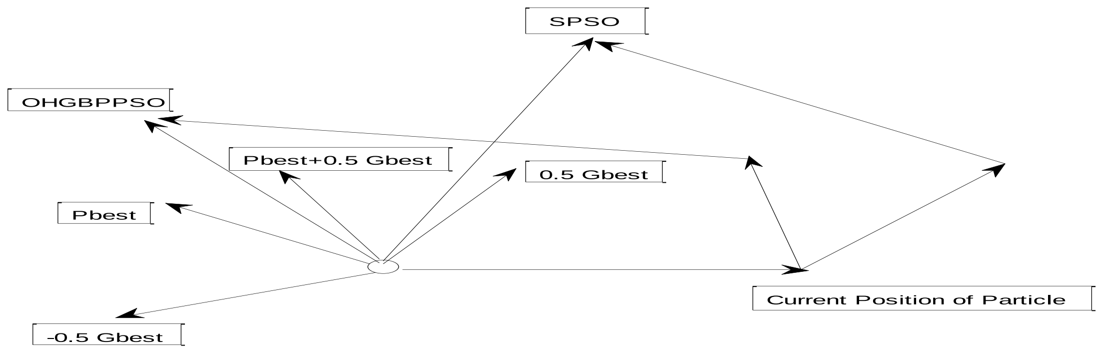


Figure-II: Comparison Movement of Particle SPSO and OHGBPPSO by Non-Scalable Problems.



3 TEST PROBLEMS

The relative performance of SPSO and OHGBPPSO is evaluated on two kinds of problem sets. Problem Set 1 consists of 15 scalable problems and Problem Set-II consists of 13 non-scalable Problems.

4 SCALABLE AND NON-SCALABLE PROBLEMS

4.1 Scalable Problem: In which scalable problem the problem size is increase and decrease according to time.

4.2 Non-Scalable Problem: In which non-scalable problem the problem size is fixed, but the problems have many local as well as global optima.

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

5. PARAMETER SETTING AND ANALYSIS OF RESULTS

5.1 Parameter Setting: 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 OHGBPPSO are set to be $c_1 = c_2 = 1.5$.

5.2 Results Analysis: In observing Table 3, it can be seen that OHGBPPSO gives a better quality of solutions as compared to SPSO. Thus, for the scalable problems OHGBPPSO outperforms SPSO with respect to efficiency, reliability, cost and robustness. In observing Table 4, it can be seen that OHGBPPSO gives a better quality of solutions as compared to SPSO. Thus, for the non-scalable problems OHGBPPSO outperforms SPSO with respect to efficiency, reliability, cost and robustness, Table 3.

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

Table-3 Comparison of SPSO and OHGBPPSO by Scalable Problems Set-I

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Rate of Success	
	SPSO	OHGBPPSO	SPSO	OHGBPPSO	SPSO	OHGBPPSO	SPSO	OHGBPPSO
1	0.674207	0.524363	14699.6000	3835.60000	0.323300	0.089916	**68.00%**	100%
2	0.683359	0.415349	902.400000	812.400000	0.051324	0.109088	100%	100%
3	0.000000	0.000000	40.000000	40.000000	0.000569	0.000483	100%	100%
4	0.770386	0.680129	7038.80000	4505.20000	0.024076	0.053167	100%	100%

5	20.89413	0.222967	13000.0000	5724.00000	16.219312	0.393235	**0.00%**	100%
6	0.007444	0.002482	140.400000	130.000000	0.263601	0.280204	100%	100%
7	0.001967	0.000010	52.800000	64.400000	0.251745	0.218062	100%	100%
8	0.000000	0.000000	40.000000	40.000000	0.029778	0.025274	100%	100%
9	0.000005	0.000004	40.400000	44.400000	0.135186	0.253899	100%	100%
10	0.001648	0.001330	40.400000	42.000000	0.141846	0.225886	100%	100%
11	0.635139	0.172111	5653.20000	4446.40000	0.065082	0.189287	100%	100%
12	0.012020	0.011124	65.200000	74.000000	0.256712	0.235492	100%	100%
13	0.057514	0.047514	1196.00000	1587.00000	0.216409	0.246410	100%	100%
14	0.002002	0.002000	40.000000	40.000000	0.129936	0.147956	100%	100%
15	0.000000	0.000000	40.000000	40.000000	0.011661	0.013672	100%	100%

Table-4 Comparison of SPSO and OHGBPPSO by Non-Scalable Problems Set-II

Problem No.	Minimum Function Value		Mean Function Value		Standard Deviation		Success of Rate	
	SPSO	OHGBP PSO	SPSO	OHGBP PSO	SPSO	OHGBP PSO	SPSO	OHGBPPSO
1	0.500000	0.500104	41.600000	49.600000	0.088031	0.089541	100%	100%
2	0.004665	0.015410	48.800000	50.000000	0.279521	0.251598	100%	100%
3	0.002060	0.009141	54.400000	61.200000	0.240645	0.239715	100%	100%
4	0.003335	0.006881	81.600000	86.400000	0.257595	0.258021	100%	100%
5	0.002562	0.034907	60.400000	59.600000	0.249756	0.251556	100%	100%
6	73046.5964	73046.59648	30000.000	30000.000	0.000000	0.000000	0.00%	0.00%
7	14.541432	26.900800	30000.000	30000.000	0.000000	0.000000	0.00%	0.00%
8	0.009239	0.091784	59.200000	96.800000	0.286622	0.230102	100%	100%
9	0.480964	0.480470	40.000000	40.000000	0.037841	0.033635	100%	100%
10	0.075842	0.034330	578.80000	546.80000	0.233322	0.245201	100%	100%
11	0.006245	0.006541	46.800000	49.600000	0.206116	0.236984	100%	100%
12	0.017029	0.003487	51.200000	56.800000	0.262820	0.229284	100%	100%
13	0.010104	0.012869	45.600000	54.400000	0.217098	0.210105	100%	100%

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

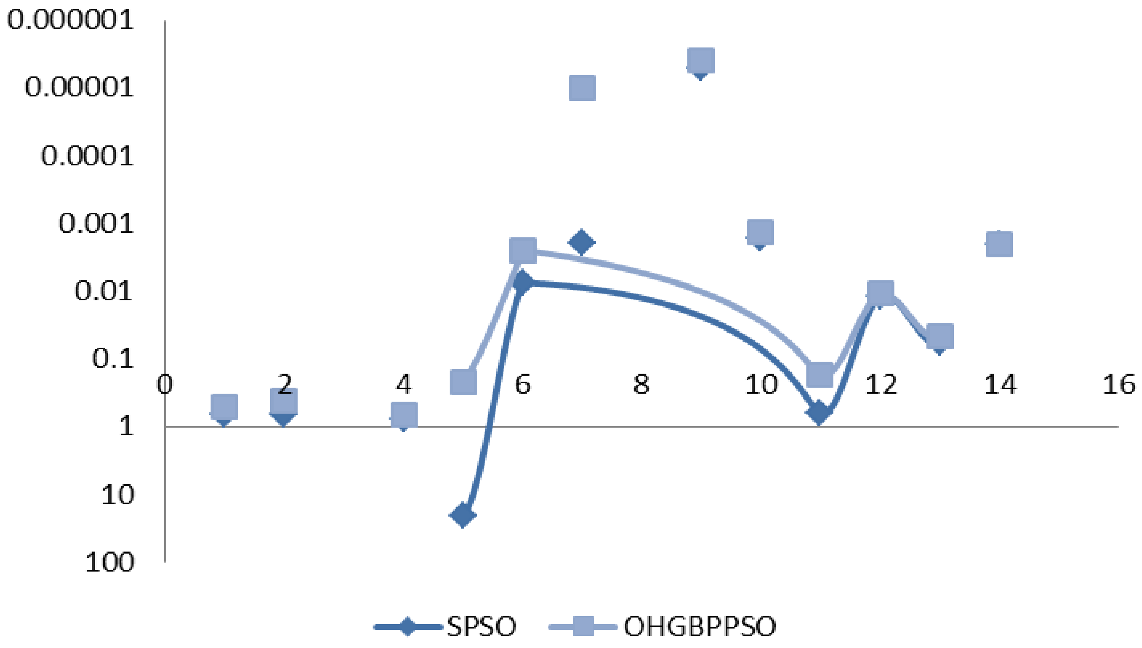
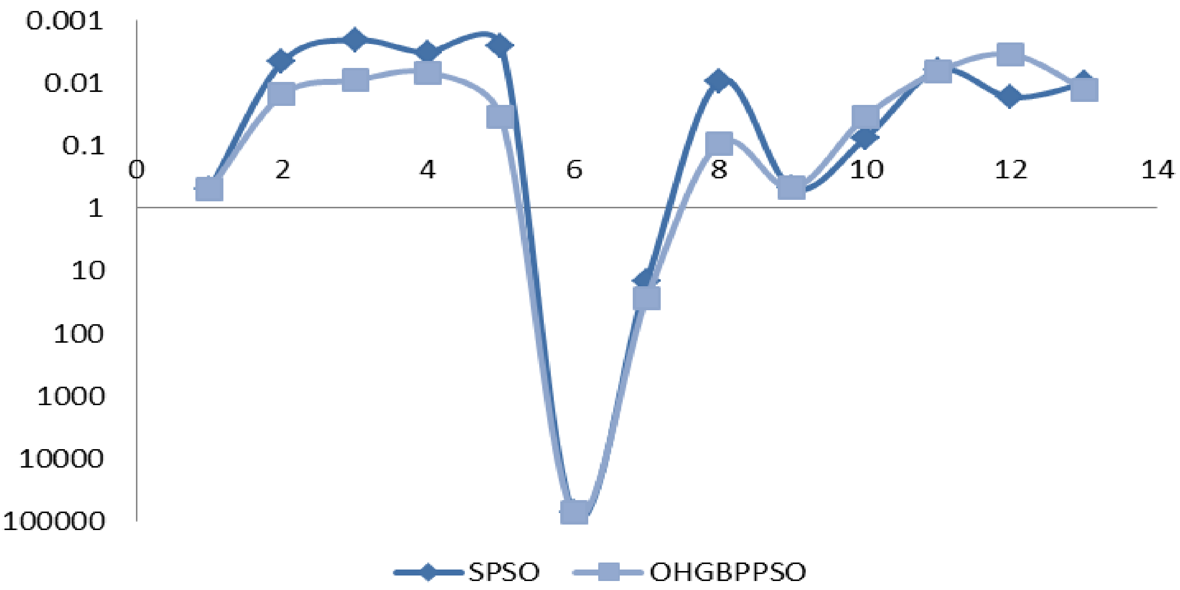


Figure B: Comparing the SPSO and OHGBPPSO with the help of 13 Non-Scalable Problems SET-II



Note: x-axis represented the scalable and non-scalable problems and y-axis denoted the Minimimum Objective Function Values

6 CONCLUSIONS

In this paper, a new PSO approach One Half Global Best Position Particle Swarm Optimization is presented. The algorithm is tested on scalable problems (increasing or decreasing particle in the swarm) and non-scalable problems (swarm size is fixed). The results show that 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.

REFERENCES

[1] R.C. Eberhart and J. Kennedy A New Optimizer using Particle Swarm Theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, 1995, pp 39–43.

[2] J. Kennedy and R.C. Eberhart. Particle Swarm Optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1995, pp 1942–1948. IEEE Press.

[3] J. Kennedy. Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, July 1999, pages 1931–1938.

[4] J. Kennedy and R. Mendes Population Structure and Particle Performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002. pages 1671–1676. IEEE Press.

[5] E.S. Peer, F. van den Bergh, and A.P. Engelbrecht. Using Neighborhoods with the Guaranteed Convergence PSO. In *Proceedings of the IEEE Swarm Intelligence Symposium*, 2003, pp 235–242. IEEE Press.

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

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

[8] F. van den Bergh *An Analysis of Particle Swarm Optimizers*. PhD thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[9] F. van den Bergh and A.P. Engelbrecht. A Study of Particle Swarm Optimization Particle Trajectories. *Information Sciences*, 176(8) , 2006, pp 937–971.

[10] J. Kennedy. Bare Bones Particle Swarms. In *Proceedings of the IEEE Swarm Intelligence Symposium*, April 2003, pp 80–87.

[11] Y. Shi and R.C. Eberhart. A Modified Particle Swarm Optimizer. In *Proceedings of the IEEE Congress on Evolutionary Computation*, May 1998, pp 69–73.

[12] Angline, P.J. 'Evolutionary optimization versus particle swarm optimization philosophy and performance differences', *Lecture Notes in Computer Science*, Vol.1447, 1998a pp.601-610, Springer, Berlin.

[13] Angline, P.J 'Using selection to improve particle swarm optimization', Proceedings of the IEEE Conference on Evolutionary Computations, 1998b pp.84-89.

[14] Clerc M., Kennedy J., " The Particle Swarm : Explosion, Stability, and Convergence in a Multi-dimensional Complex Space", IEEE Transactions on Evolutionary Computation,Vol.6, 2002, pp 58-73.

[15] Eberhart, R. C. and Shi, Y. Comparing inertia weigthts and constriction factors in particle swarm optimization. *Proceedings of IEEE Congress on Evolutionary Computation, 2000* pp. 84-88. San Diego, CA.

[16] Z-H. Zhan, J. Zhang, Y. Li, and H.S-H. Chung. Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, 2009, pp 1362-1381.

[17] Z. Xinchao. A perturbed particle swarm algorithm for numerical optimiza-tion. Applied Soft Computing, 2010, pp 119-124.

[18] T. Niknam and B. Amiri. An efficient hybrid approach based on PSO,ACO and k-means for cluster analysis. Applied Soft Computing, 2010, pp 183- 197.

[19] M. El-Abda, H. Hassan, M. Anisa, M.S. Kamela, and M. Elmasry. Discrete cooperative particle swarm optimization for FPGA placement. Applied Soft Computing, 2010 pp 284-295.

[20] M-R. Chena, X. Lia, X. Zhanga, and Y-Z. Lu. A novel particle swarm optimizer hybridized with extremal optimization. Applied Soft Computing, 2010, pp 367-373.

[21] P.W.M. Tsang, T.Y.F. Yuena, and W.C. Situ. Enhanced a_ne invariant matching of broken boundaries based on particle swarm optimization and the dynamic migrant principle. Applied Soft Computing, 2010, pp 432-438.

[22] C-C. Hsua, W-Y. Shiehb, and C-H. Gao. Digital redesign of uncertain interval systems based on extremal gain/phase margins via a hybrid particle swarm optimizer. Applied Soft Computing, 2010, pp 606-612.

[23] H. Liua, Z. Caia, and Y. Wang. Hybridizing particle swarm Optimi-

-zation with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing, 2010, pp 629-640.

[24] K. Mahadevana and P.S. Kannan. Comprehensive learning particle swarm optimization for reactive power dispatch. Applied Soft Computing, 2010, pp 641-652.

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

[26] M.E.H. Pedersen and A.J. Chipper_eld. Simplifying particle swarm optimization. Applied Soft Computing, 2010, pp 618-628.