Learning in Particle Swarm Optimization

Shailendra S. Aote, M.M.Raghuwanshi, L.G.Malik

Abstract— **This paper presents particle swarm optimization based on learning from winner particle. (PSO-WS). Instead of considering gbest and pbest particle for position update, each particle considers its distance from immediate winner to update its position. Only winner particle follow general velocity and position update equation. If this strategy performs well for the particle, then that particle updates its position based on this strategy, otherwise its position is replaced by its immediate winner particle's position. Dimension dependant swarm size is used for better exploration. Proposed method is compared with CSO and CCPSO2, which are available to solve large scale optimization problems. Statistical results show that proposed method performs well for separable as well as non separable problems.**

Index Terms— **Large Scale Optimization, Particle Swarm Optimization, Winners Strategy.**

I. INTRODUCTION

As many researchers are attracted towards the field of optimization, many optimization methods starting from single objective optimization to multi-objective optimization, unimodal optimization to multimodal optimization, are proposed in past years. Optimization means problem solving in which one seeks to minimize or maximize the objective function in presence of the constraints. Difficulty in solving the problem increases with the increase in dimension size.

Particle swarm optimization (PSO) is global optimization method proposed initially by James Kennedy and Russell Eberhart [1] in 1995. The particle swarm concept originated as a simulation of simplified social system. When a flock of birds travel in the air to find the food, they maintain their velocity and position, so that they never collide each other. There is no central coordinator for the movement of flock. Movement takes place based on cooperation among each bird. Due to its simple equation and involvement of less parameter, it becomes one of the promising algorithms for solving optimization problems.

Each particle is treated as a point in a D dimensional space. The ith particle is represented as

Xi = (Xi1, Xi2,…………,XiD)

The best previous position (the position giving the best fitness value) of any particle is recorded and represented as

Pi = (Pi1, Pi2,……..,PiD)

The index of the best particle among all the particles in the population is represented by the symbol g. The rate of the position change (velocity) for particle i is represented as

Vi = (Vi1,Vi2,……….,ViD)

The velocity and position of the particles are updated according to the following equation,

Vid=Vid+c1*rand()*(Pid-Xid)+c2*rand()*(Pgd-Xid) (1)

Xid =Xid + Vid (2)

rand() is a random function gives value in the range[0,1]. The first part of the equation (1) talks about previous velocity; whereas the second part is the "cognition' part that represents the private thinking of the particle. The third part is the "social" part that represents the collaboration among the particles. Therefore, equation (1) is calculate the particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience. Particle moves toward a new position according to equation (2). The performance of each particle is measured according to a predefined fitness function, which is related to the problem to be solved.

These basic PSO equations are updated by researchers. The inertia weight w is brought into the equation (1) as shown in equation (3) by Yuhui Shi in 1998[8]. This w helps in balancing the global search and local search. It can be a positive constant or even a positive linear or nonlinear function of time.

Vid=w*Vid+c1*rand()*(Pid-Xid)+c2*rand()*(Pgd-Xid) (3)

Initially value 0.7 was proposed for w that is further considered as dynamic for better exploration in the beginning and for fine exploitation at the end. Lots of PSO variants are proposed from its formation to till date. Efforts are put towards increase in efficiency and convergence speed. Both the things are rarely achieved in a single algorithm. Low dimensional problems are easier to solve, where as complexity increase as the increase in dimensions and modality. A niching method is introduced in EAs to locate multiple optimal solutions [2]. Distance based LIPS model [3], a memetic PSO [4], Adaptive PSO [5], Fractional PSO [6], AGPSO [7], CSO [33] and many other techniques are proposed to handle higher dimensional and multimodal problems. In spite of these techniques, trapping in local minima and the rate of convergence are two unavoidable problems in PSO and all other EAs. George I. Evers proposed a RegPSO [13], where problem of stagnation is removed by automatically triggering the swarm regrouping. Efforts are taken to solve multimodal problems. To solve higher dimensional problems, variants cooperative co evolution strategies like CPSO-SK and CPSO-HK [14], CCPSO [15], CCPSO2 [16] are proposed.

Following the idea in [33], where losers are updated base on winners. A pair wise is competition is takes place between the particles. Different pairs among the particle are randomly formed. The problem in this strategy is that the loser in one pair may be winner for other pair. So, in this paper, instead of forming the pair randomly, we introduced the concept of immediate winner for each particle. Each particle will update the position based on immediate winners.

Rest of the paper is organized as follows. Section II represents the PSO with Learning. Experimental results and analysis is given into Section III, followed by Conclusion in section IV.

Shailendra S. Aote, *Research Scholar, Dept. of CSE, GHRCE Nagpur, India, Mobile N: 9096896565*
M.M.Raghuwanshi*, Professor, Dept of CT, YCCE, Nagpur*
L.G.Malik*, Professor, Dept of CSE, GHRCE, Nagpur*

II. PSO WITH LEARNING

In order to solve optimization problems, a new PSO with winner strategy is defined here. General PSO considers gbest and pbest to update the position in each iteration. Instead of considering pbest and gbest, we consider immediate winner particle for each. In each iteration, it first evaluates the swarm to calculate the fitness. Sort the swarm in increasing order of the fitness. The best particle follow the general velocity and position update equation, where the third part of velocity equation is absent because, the best particle itself is a gbest. Other particles consider their immediate winner particle for position update. Immediate winner is termed as given: if i is a current position, then i-1 is considered as winner position. This strategy is more promising as no randomness is involved to define winner as given in [33]. The velocity update equation for particle i is,

vel(i) = w*vel(i) + Φ * rand* (vel(i-1) – vel(i))

Where Φ is clerk's coefficient that controls the swarm and w is the inertia value, which is ranging from 0.9 to 0.4. If the new position of the particle is better, that particle enters into next generation otherwise its position is replaced by its winner particle's position.

The proposed algorithm is as follows

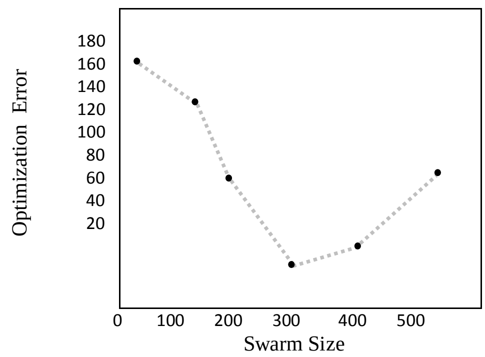
START
Swarm size = Number of Dimensions
Initialize position and velocity the swarm.
fit = Evaluate the swarm.
While *termination criteria* is not satisfied **do**
Sort the swarm
The best particle follows same velocity and position update equation as per eq. (3) and (2).
for j = 2 to size of swarm
vel(j) = w*vel(j) + Φ * rand* (vel(j-1) – vel(j))
pos(j) = pos (j) + vel (j)
end ***for***
fit1= Evaluate the swarm
for j = 2 to size of swarm
if (fit1(j) < fit (j)
Update the position of particle
else
pos(j) =pos (j-1);
end ***if***
end ***for***
end ***While***
End ***Start***

Computational complexity of PSO-WS is *O(mn)*, where m is swarm size and n is the search dimensions.

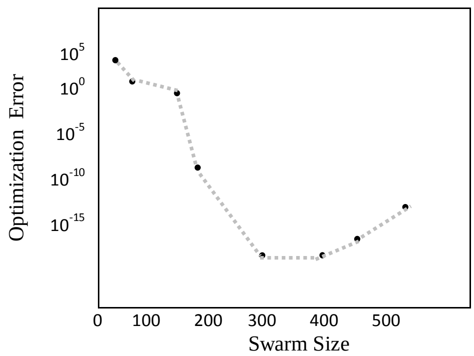
III. EXPERIMENTAL RESULTS & ANALYSIS

Proper parameter setting plays very important role in solving the optimization problem. One of the most important parameter is the swarm size. With the small swarm size, convergence is very fast but may cause premature convergence without exploring the search space. It doesn't mean that swarm size is large, which may cause slower convergence speed. Fig 1 shows the performance of f2 and f6 to decide the swarm size for 1000 dimensions.

50 independent runs are taken for each function and its mean, median, standard deviation and best value is noted for comparison. For each independent run, the maximum number of FEs, as recommended in, is set to $5000 * n$, where n is the search dimension of the test functions. In the comparisons between different statistical results, two-tailed *t*-tests are conducted at a significance level of $\alpha = 0.05$. The performance is compared with CSO and CCPSO-2, which are known as the state of the art algorithm for large scale optimization problems. Performance is evaluated on 100, 500 and 1000 dimensions on CEC-08 test suit [32] and shown the in table I, II and III respectively



a) f2



b) f6

Fig: Optimization error obtain by PSO-WS on (a) *f2* and (b) *f6* of 1000 dimensions with different swarm sizes varying from 25 to 500

Results of CSO & CCPSO2 are taken from [33]. Values outside the brackets show the mean performance, whereas values inside the first brackets show standard deviation. Values for CSO and CCPSO2 in second bracket shows p-value. Unpaired t-test was conducted between PSO-WS and CSO also between PSO-WS and CCPSO2. If two results are statistically significant shown in bold, then the comparision takes place. Results of test is shown in w/l/NC form, where w,l,NC stands for winning count, loser count and no conclusion count. No conclusion is made for Statistically different results.

TABLE I. COMPARISON OF PSO-WS WITH CSO AND CCPSO2 ON 100 DIMENSIONS

100D	**PSO-WS**	**CSO**	**CCPSO2**
$CEC08_1$	7.28E-33(1.87E-33)	9.11E-29(1.10E-28) **(<0.0001)**	7.73E-14 (3.23E-14) **(0.0176)**
$CEC08_2$	6.12E+00(8.01E+00)	3.35E+01(5.38E+00) **(<0.0001)**	6.08E+00 (7.83E+00)(0.9972)
$CEC08_3$	1.25E+00(3.11E+00)	3.90E+02(5.53E+02) **(<0.0001)**	4.23E+02 (8.65E+02)(0.626)
$CEC08_4$	2.09E-05(1.17E-05)	5.60E+01(7.48E+00) **(<0.0001)**	3.98E-02 (1.99E-01)(0.841)
$CEC08_5$	7.20E-05(9.19E-05)	0.00E+00(0.00E+00) (0.4343)	3.45E-03 (4.88E-03)(0.489)
$CEC08_6$	4.61E-16(9.34E-16)	1.20E-014(1.52E-015)	1.44E-13 (3.06E-14) **(<0.0001)**
$CEC08_7$	-1.93E+01(1.14E+00)	-7.25E+05(1.88E+04)	-1.50E+03 (1.04E+01) **(<0.0001)**
w / l / NC for mean		6 / 0 / 1	3 / 0 / 4
w / l / NC for Std. Deviation		5 / 1 / 1	3 / 0 / 4

TABLE II. COMPARISON OF PSO-WS WITH CSO AND CCPSO2 ON 500 DIMENSIONS

500D	**PSO-WS**	**CSO**	**CCPSO2**
$CEC08_1$	9.28E-28(1.29E-29)	6.57E-23(3.90E-24) **(<0.0001)**	7.73E-14 (3.23E-14)**(0.009)**
$CEC08_2$	9.15E+00(3.21E+00)	2.60E+01(2.40E+00) **(<0.0001)**	5.79E+01 (4.21E+01)(0.207)
$CEC08_3$	3.51E+01(1.17E+01)	5.74E+02(1.67E+02)**(0.0005)**	7.24E+02 (1.54E+02) **(<0.0001)**
$CEC08_4$	4.29E-04(8.21E-02)	3.19E+02(2.16E+01) **(<0.0001)**	3.98E-02 (1.99E-01)(0.8461)
$CEC08_5$	1.93E-20(5.01E-10)	2.22E-16(0.00E+00)(1.000)	1.18E-03 (4.61E-03)(0.7792)
$CEC08_6$	5.81E-15(8.21E-15)	4.13E-13(1.10E-014) **(<0.0001)**	5.34E-13 (8.61E-14) **(<0.0001)**
$CEC08_7$	-7.69E+03(1.61E+01	-1.97E+06(1.88E+04) **(<0.0001)**	-7.23E+03 (4.16E+01)
w / l / NC for mean		6 / 0 / 1	4 / 0 / 3
w / l / NC for Std. Deviation		5 / 1 / 1	4 / 0 / 3

TABLE III. COMPARISON OF PSO-WS WITH CSO AND CCPSO2 ON 1000 DIMENSIONS

1000D	**PSO-WS**	**CSO**	**CCPSO2**
$CEC08_1$	7.08E-26(3.96E-2	1.09E-21(4.20E-23) **(<0.0001)**	5.18E-13(9.61E-14) **(<0.0001)**
$CEC08_2$	4.13E+01(4.21E+	4.15E+01(9.74E-01)(0.9631)	7.82E+01(4.25E+01) (0.3878)
$CEC08_3$	6.97E+02(6.31E+	1.01E+03(3.02E+01) **(<0.0001)**	1.33E+03(2.63E+02) **(0.0195)**
$CEC08_4$	9.73E-03(6.84E-0	6.98E+02(3.10E+01) **(<0.0001)**	1.99E-01(4.06E-01) (0.6458)
$CEC08_5$	3.85E-19(6.33E-1	2.26E-16(2.28E-17) **(<0.0001)**	1.18E-03 (3.27E-03) (0.7183)
$CEC08_6$	7.34E-14(4.23E-1	1.21E-12(2.64E-014) **(<0.0001)**	1.02E-12 (1.68E-13) **(<0.0001)**
$CEC08_7$	-9.18E+03(8.12E	-3.83E+06(4.82E+04) **(<0.0001)**	-1.43E+04(8.27E+01) **(<0.0001)**
w / l / NC for mean		6 / 0 / 1	4 / 0 / 3
w / l / NC for Std. Deviation		5 / 1 / 1	4 / 0 / 3

RESULTS SHOW THAT PSO-WS OUTPERFORMS ON MEAN AS WELL AS STANDARD DEVIATION FOR MOST OF THE FUNCTION, WHERE STATISTICAL RESULTS ARE SIGNIFICANT.

IV. CONCLUSION & FUTURE DIRECTION

In this paper, a new strategy based on immediate winner particle is proposed. After sorting the fitness of particles, each particle updates its position, based on the particle, which has better fitness value. If new position is better as compared to previous position, then it is accepted, otherwise its position is replaced with winner particle's position. To have a better exploration, dimension dependant swarm size is used. The proposed algorithm is compared with two state of the art algorithm for large scale optimization known as CSO and CCPSO2 on CEC-08 test suit. Statistical results show that proposed method performs extremely well on most of the functions.

Future work is to provide theoretical proof of convergence for the proposed method. To test the scalability, proposed Algorithm can be implemented up to 5000 dimensions. This method can be used to solve multiobjective optimization problems.

REFERENCES

[1] James Kennedy and Russell Eberhart , "Particle Swarm Optimization", IEEE, International Conference on Neural Network, Perth, Australia1995, pp. 1942-1948.

[2] Xiaodong Li and Kalyanmoy Deb, "Comparing lbest PSO Niching algorithms Using Different PositionUpdate Rules", WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain, pp. 1564-1571.

[3] B. Y. Qu, P. N. Suganthan and Swagatam Das, "A Distance-based Locally Informed Particle Swarm Model for Multi-modal Optimization" , IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 17, NO. 3, JUNE 2013, pp .387-402.

[4] Hongfeng Wang, Ilkyeong Moon, Shenxiang Yang ,Dingwei Wang, "A memetic particle swarm optimization algorithm for multimodal optimization problems", Information Sciences 197 (2012) pp. 38–52.

[5] Zhi-Hui Zhan, Jun Zhang, Yun Li and Henry Shu-Hung Chung, "Adaptive Particle Swarm Optimization", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 6, DECEMBER 2009, pp 1362-1381.

[6] Serkan Kiranyaz, Turker Ince, AlperYildirim, and Moncef Gabbouj, "Fractional Particle Swarm Optimization in Multidimensional Search Space", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 40, No. 2, April 2010, Pp 298-319.

[7] Seyedali Mirjalili, Andrew Lewis, Ali Safa Sadiq, "Autonomous Particles Groups for Particle Swarm Optimization", Arab J Sci Eng (2014) 39, pp.4683–4697

[8] Yuhui Shi and Russell Eberhart, "A Modified Particle Swarm Optimizer", proceedings of the IEEE congress on Evolutionary Computation(CEC 1998), Piscataway, NJ. 69-73, 1998

[9]F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D.dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa,2002.

[10] H. Wang, et al., "Opposition-based particle swarm algorithm with Cauchy mutation," in Proceedings ofthe IEEE Congress on Evolutionary Computation,2007, pp. 4750-4756.

[11] Shuyuan Yang, Min Wang, Licheng Jiao, “A Quantum Particle Swarm Optimization”, proceedings of the IEEE congress on Evolutionary Computation(CEC 2004), pp. 320 – 324.

[12] Stefan Janson and Martin Middendorf, “A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant”, IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 35, No. 6, December 2005, Pp. 1272- 1282.

[13] George I. Evers and Mounir Ben Ghalia, “Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks”, IEEE International Conference on Systems, Man and Cybernetics, 2009.pp 3901-3908.

[14] F. van den Bergh and A. Engelbrecht, “A cooperative approach to particle swarm optimization,” IEEE Trans. Evol.Computation., vol. 8, no. 3, pp. 225–239, Jun. 2004.

[15] Z. Yang, K. Tang, and X. Yao, “Large scale evolutionary optimization using cooperative coevolution,” Inform. Sci., vol. 178, no. 15, pp.2986–2999,Aug. 2008.

[16]Cui, Z.; Zeng, J.; Yin, Y.: An improved PSO with time-varying accelerator coefficients. In: Eighth international conference on intelligent systems design and applications, Kaohsiung, pp 638–643 (2008)

[17]Ziyu T, Dingxue Z., “:A modified particle swarm optimization with an adaptive acceleration coefficients”, Asia-Pacific Conference on Information Processing, Shenzhen, pp. 330–332 (2009)

[18] Bao, G.Q.; Mao, K.F.: Particle swarm optimization algorithmwith asymmetric time varying acceleration coefficients. In: IEEE International Conference on Robotics and Biomimetics, Guilin, 2009, pp. 2134–2139.

[19] Yuntao Dai, Liqiang Liu, Ying Li, “An Intelligent Parameter Selection Method for Particle Swarm Optimization Algorithm”, Fourth International Joint Conference on Computational Sciences and Optimization, 2011, pp.960-964.

[20] M. Clerc and J. Kennedy, “The particle swarm: Explosion, stability, and convergence in a multidimensional complex space,” IEEE Trans. Evol. Comput., vol. 6, no. 1, pp. 58–73, Feb. 2002.

[21] T. Ray and X. Yao, “A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning,” in Proc. IEEE CEC, May 2009, pp. 983–999.

[22] S. Zhao, J. Liang, and P. Suganthan, “Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization,” in Proc. IEEE CEC, Jun. 2008, pp. 3845–3852.

[23] XianjunShen, Zhifeng Chi, Jincai Yang, Caixia Chen, Zhifeng Chi, “Particle Swarm Optimization with Dynamic Adaptive Inertia Weight”, International Conference on Challenges in Environmental Science and Engineering,2010, pp 287-289.

[24] Sabine Helwig, JuergenBranke, Member, IEEE, and Sanaz Mostaghim, “Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization”, IEEE, 2011.

[25] Mohammad Nabi Omidvar, and Xiaodong Li, Xin Yao, “Cooperative Co-evolution with Delta Grouping for Large Scale Non-separable Function Optimization”, WCCI 2010 IEEE World Congress on Computational Intelligence, July, 18-23, 2010 - CCIB, Barcelona, Spain, pp. 1762-1769

[26] M.G. Epitropakisa,c,, V.P. Plagianakosb,c, M.N. Vrahatis, “Evolving cognitive and social experience in Particle Swarm Optimization through Differential Evolution: A hybrid approach”, Information Sciences 216 (2012,) pp. 50–92.

[27] Ratnaweera A, Halgamuge SK, and Watson HC, “Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients”,IEEE Transactions on Evolutionary Computation, VOL. 8, NO. 3, JUNE 2004, pp. 240 – 255.

[28] Y. Shi and R. C. Eberhart ,“Empirical study of particle swarm optimization,” in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 3, 1999, pp. 101–106.

[29] Bonyadi, Mohammad Reza, Zbigniew Michalewicz, and XiaodongLi,"An analysis of the velocity updating rule of the particle swarm optimization algorithm." Journal of Heuristics 20.4 (2014): 417-452.

[30] Xiaodong Li, Xin Yao, “Cooperatively Coevolving Particle Swarms for Large Scale Optimization”, IEEE Transactions On Evolutionarycomputation,Vol.16,No.2,April2012,Pp.210-224.

[31] Bonyadi, Mohammad Reza, and Zbigniew Michalewicz. "A locally convergent rotationally invariant particle swarm optimization algorithm."Swarm Intelligence 8.3 (2014): 159-198.

[32] K. Tang, X. Yao, P. Suganthan, C. MacNish, Y. Chen, C. Chen, and Z. Yang, “Benchmark functions for the CEC’2008 special session and competition on large scale global optimization,” Nature Inspired Computat. Applicat. Lab., Univ. Sci. Technol. China, Hefei, China, Tech,rep. 2007 [Online]. Available: http://nical.ustc.edu.cn/cec08ss.php.

[33] Ran Cheng and Yaochu Jin, “A Competitive Swarm Optimizer for Large Scale Optimization”, IEEE TRANSACTIONS ON CYBERNETICS, VOL. 45, NO. 2, FEBRUARY 2015, Pp 191-204.

