Impact of Variable Links on Standard Particle Swarm Optimization

Mrs Snehal Mohan Kamalapur
Research Scholar, Dr. D. Y. Patil Institute of Engineering & Technology, Pune, Savitribai Phule P
une University, India
snehal_kamalapur@yahoo.com

Dr. Varsha Hemant Patil
Supervisor, Dr. D. Y. Patil Institute of Engineering & Technology, Pune, Savitribai Phule Pune University, India
varsha.patil@gmail .com

Abstract—Particle Swarm Optimization (PSO) is inspired by collective behavior in the nature and social sharing of information of a population is the core idea behind PSO. Researchers have proposed different variants of PSO to verify the effects of various parameters on the performance of PSO. PSO algorithm has gained increasing interest in recent years to deal with optimization problems. Clerc proposed Standard PSO 2011(SPSO2011) version which modifies velocity in a geometrical way as early PSO variants were rotationally variant. Proposed work focuses on impact of variable number of links for different threshold number of unsuccessful iterations. Benchmark suite used here is CEC2013. As particles in the swarm may leave the feasible search space, different bound handling methods have been proposed so far. The work here considers reflex position bound handling with hyperbolic velocity bound handling.

***Keywords** - Bound Handling Methods, Particle Swarm Optimization, Standard PSO, Variable Link*

*****

I. INTRODUCTION

Particle swarm optimization (PSO) is optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995[1][2]. It is one of the most important swarm intelligence paradigms [3]. It is inspired by social behavior patterns of organisms that live and interact within large groups like fish and birds. It has been applied to solve many real-world optimization problems. PSO is population based and has swarm of particles which represent potential solution.

PSO has very few parameters [4-6] like swarm size, inertia weight [7-18], acceleration coefficient [19-22 and neighborhood topology to set. Neighborhood topologies determine how particles communicate with each other. Star, Ring, Mesh, Tree, Ring Lattice with dynamic increment neighborhood [23], Fitness Distance Ratio [24], Dynamic Hierarchy [25], Adaptive Random [3] and TRIBES are neighborhood topologies. The adaptive random topology is considered here for experimentation. The particles in the swarm tend to leave the search space. Different bound handling methods are used in the literature to restrict these particles in search space. Variants proposed in this paper consider reflex position bound handling and hyperbolic velocity bound handling.

Standard PSO (SPSO) variant, reflex and hyperbolic confinement methods are briefed in Section 2 and 3 respectively. Section 4 presents experimental setup with a set of benchmark functions and variants used for testing performance of experimentation. Results are discussed in Section 5 and conclusions are drawn in Section 6.

II. STANDARD PSO

Three Standard PSO algorithms have been defined so far SPSO2006, SPSO2007 and SPSO2011. SPSO 2011 improves earlier PSO which was sensitive to the system of coordinates. Standard PSO algorithms initialize random position inside the search space for each particle.

$$X_i(0) = U(X_{min}, X_{max}) \tag{1}$$

Where $U(X_{min}, X_{max})$ is a random number drawn in $[X_{min}, X_{max}]$ according to the uniform distribution.

Previous best is set as initial position.

$$P_i(0) = X_i(0) \tag{2}$$

In SPSO 2006 and SPSO 2007 velocity is initialized as

$$V_i(0) = \frac{U(X_{min}, X_{max}) - X_i(0)}{2} \tag{3}$$

Particles leave the search space when the dimension D is high using above formula [3] . Hence SPSO 2011 uses-

$$V_i(0) = U\big(X_{min} - X_i(0), X_{max} - X_i(0)\big) \tag{4}$$

SPSO 2011 uses 40 as swarm size while for SPSO 2006 and SPSO 2007, the swarm size is determined using –

$$S = 10 + [2 * sqrt(D)] \tag{5}$$

All three versions of standard PSO use same inertia weight and acceleration coefficient value.

$$\omega = 1/\big(2\, ln(2)\big) = 0.721 \tag{6}$$

$$c = 1/2 + ln(2) = 1.193 \tag{7}$$

Clerc [3] has defined adaptive Random topology, which is used in all three versions of SPSO. Each particle informs K particles at random and informs itself. Thus each particle informs at least one particle and atmost K+1 particle. The

 ISSN: 2321-8169
278– 283

information links is modified at the beginning and after unsuccessful iteration. Generally K is set to 3. PSO finds optimum point very easily when the optimum point lies on a axis, on a diagonal, or on the centre of the system of coordinates because of dimension by dimension method. Hence SPSO 2011 updates velocity in geometrical way.

$$G_i = \frac{X_i + (X_i + c*(P_i - X_i)) + (X_i + c*(L_i - X_i))}{3} \quad (8)$$

$$G_i = X_i + c * \frac{(P_i + L_i - 2*X_i)}{3} \quad (9)$$

Let x'_i is a random point drawn in hypersphere according to the uniform distribution-

$$H_i(G_i, \|G_i - X_i\|) \quad (10)$$

$$v_{id}(t+1) = \omega * v_{id}(t) + x'_{id}(t) - x_{id}(t) \quad (11)$$

Particles may leave the search space in PSO during the iterations. The particle should stay inside $[X_{min};X_{max}]$. SPSO uses nearest for position confinement-

When the position of the particle moves away from the boundary, the particle is set back using -

If $x_{i,d}(t+1)>x_{max}$ then $x_{i,d}(t+1)=x_{max}$

If $x_{i,d}(t+1)<x_{min}$ then $x_{i,d}(t+1)=x_{min}$

SPSO uses Absorb or Zero confinement method for velocity when the position of the particle moves away from the boundary. SPSO 2011 is considered here for experimentation. Section 4 briefs different confinement methods from PSO literature.

III. BOUND HANDLING METHODS

Particles may leave the search space. To prevent particles to leave the search space position and velocity bound handling methods are proposed in the literature. Absorb or Zero, Deterministic Back, Random Back and Hyperbolic are major velocity bound handling methods while Nearest, Do not Change and Reflex are major position bound handling methods [12]. When the position of the particle moves away from the boundary the particle is set back using bound handling methods.

3.1 Reflex Position Confinement Methods

Particles may leave the search space in PSO during the iterations.

- Reflex

If $x_{i,d}(t+1) < x_{min}$ then $x_{min} + (x_{min} - x_{i,d}(t+1))$,

If $x_{i,d}(t+1) > x_{max}$ then $x_{max} - (x_{i,d}(t+1) - x_{max})$ (12)

3.2 Hyperbolic Velocity Confinement Methods

Whenever velocity goes beyond threshold bound then it is normalized as in eq. (13)

If velocity goes beyond threshold bound

$$\left.\begin{array}{ll} v_{i,d}(t+1) = \dfrac{v_{i,d}(t+1)}{1+\dfrac{v_{i,d}(t+1)}{\max(x_{i,d})-x_{i,d}(t)}} & \text{if } v_{i,d}>0 \\[2ex] v_{i,d}(t+1) = \dfrac{v_{i,d}(t+1)}{1+\dfrac{v_{i,d}(t+1)}{x_{i,d}(t)-\min(x_{i,d})}} & \text{if } v_{i,d}<0 \end{array}\right\} \quad (13)$$

IV. EXPERIMENTAL SET UP

The performance of variants given in table I are benchmarked against the set of test functions as given in table II. Experiments are carried out for 2 and 10 dimensional versions of each test function. Experiment is repeated for 20 times. Each run of the algorithm is designed to finish with either maximum number of iterations or the error value was lower than 1E-08. Maximum number of iterations used here are 7000 for two dimensional functions and 10000 for ten dimensional functions.

A. Initial population(Initialization) or starting configuration

The position for a particle in the swarm is initialized using uniform distribution along each dimension of the problem space.

B. Swarm Size

Swarm size is considered as 40.

C. Inertia Weight

Constant inertia weight equal to ω=1/(2*ln(2)) is used.

D. Velocity

Initially velocity is set using equation (4)

E. Acceleration coefficient

Acceleration coefficients c_1 and c_2 are set to 0.731.

F. Neighbourhood Topologies

In PSO, each particle has communication neighbourhood. Adaptive random link neighbourhood topologies are used –

Adaptive Random Link:

Random topology has been defined in [3]informs K particles in swarm. Particles are chosen randomly. The information links that is K particles are selected at the beginning, and after each unsuccessful iteration. Generally K is set to 3. Other variations of adaptive random link are tested here. Instead of changing informants after every unsuccessful iteration, threshold of unsuccessful iterations is set to change informants also by changing K =4.

G. Proposed Variants

Following variants are tested-

Table I - Variants

Variants	Threshold number of Unsuccessful iterations	Random Link (K)
Variant 1	1	3
Variant 2	10	3
Variant 3	20	3
Variant 4	1	4
Variant 5	10	4
Variant 6	20	4

H. Bound Handling Methods

Reflex bound handling method for position confinement and hyperbolic for velocity confinement are used here.

I. Benchmark Functions

Benchmark functions (CEC2013) [26] listed in table II, with two and ten dimensions are used for experimentation

Table II - Benchmark Functions

Type of Function	No.	Function	Optimum Function Value
Unimodal Functions	1.	Sphere	-1400
	2.	Rotated High Conditioned Elliptic Function	-1300
	3.	Rotated Bent Cigar Function	-1200
	4.	Rotated Discus Function	-1100
	5.	Different Powers Function	-1000
Multimodal Functions	6.	Rotated Rosenbrock's Function	-900
	7.	Rotated Schaffers F7 Function	-800
	8.	Rotated Ackley's Function	-700
	9.	Rotated Weierstrass Function	-600
	10.	Rotated Griewank's Function	-500
Composition Functions	11.	Composition Function 1 (n=5,Rotated)	700
	12.	Composition Function 2 (n=3,Unrotated)	800
	13.	Composition Function 3 (n=3,Rotated)	900
	14.	Composition Function 4 (n=3,Rotated)	1000

V. Results and Discussion

SPSO variants shown in table II are tested on benchmark functions. Performance is tested based on the optimum value for the function. Benchmark functions are tested for 2 and 10 dimensions. 20 runs are taken. The results is shown in Table III and IV. Bold faced entries in table III and IV indicates best performance for that function. Table V shows overall performance evaluation of all variants.

Variant3 gives 100% best performance for two dimensional functions. Variant 3 and variant 4 both gives better performance than other variants for ten dimensional functions.

SPSO2011 uses K =3 and after every unsuccessful iteration the link is modified.

Table V – Overall Performance summary of variants

Function	Best Performance Variants(No. of Dimensions of Functions Dimension -2)	Best Performance Variants (No. of Dimensions of Functions -10)
1.	All	All
2.	Variant 3	Variant 1
3.	All	Variant 3
4.	Except Variant 1	Variant 2
5.	All	All
6.	All	Variant 4
7.	All	Variant 4
8.	All	Variant 2
9.	All	Variant 3
10.	Variant 3	Variant 4
11.	All	All
12.	Variant 2 and Variant 3	Variant 2
13.	Variant 2, Variant 3, Variant 5 Variant 6	Variant 3
14.	Variant 2, Variant 3, Variant 5 Variant 6	Variant 1

VI. Conclusion

SPSO2011 updates velocity in geometrical way to make it rotationally invariant. Hence SPSO2011 is modified here. SPSO2011 uses K as 3 and the link is modified after unsuccessful iteration. The work here tests modified SPSO2011. The variants tested here use K as 3 and 4. The link is modified after threshold number 1, 10 and 20 of unsuccessful iterations. The link is modified only after threshold number of unsuccessful iterations. The results show that when K is 3 threshold 20 gives better performance for both two and ten dimensional functions. As number of dimension increases performance varies hence in future scope it can be tested for higher dimensional data.

Acknowledgement

The authors wish to thank the Principal and Research Coordinator of Padamashree Dr. D. Y. Patil Institute of Engineering and Technology, Savitribai Phule Pune University Pune.

REFERENCES

[1] Kennedy, J., and Eberhart, R. C. (1995). Particle Swarm Optimization, Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948

[2] J. Kennedy, R. C. Eberhart, and Y. H. Shi, Swarm Intelligence.San Mateo, CA: Morgan Kaufmann, 2001.

[3] Maurice Clerc. Particle Swarm Optimization. ISTE (International Scientific and Technical Encyclopedia), 2006

[4] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", in Proc.IEEE World Congr. Comput. Intell., 1998, pp. 69-73.

[5] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization",in Proc. IEEE Congr. Evol. Comput., 1999, pp. 1945-1950.

[6] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization",in Proc IEEE Congr. Evol. Comput., 2001, vol. 1, pp. 101-106.

[7] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization", Comput. Oper. Res.,vol. 33, no. 3, pp. 859-871, Mar. 2004.

[8] Y. Zheng, et. al., Empirical study of particle swarm optimizer with an increasing inertia weight, Proceeding of the IEEE Congress on Evolutionary Computation, 2003

[9] Wudai Liao, Junyan Wang, and Jiangfeng Wang, Nonlinear, Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization ICSI 2011, Part I, LNCS 6728, pp. 80- 85

[10] Huailiang Liu, Ruijuan Su, Ying Gao , Ruoning Xu, Improved Particle Swarm Optimization Using Two Novel Parallel Inertia Weights,Second International Conference on Intelligent Computation Technology and Automation, IEEE 2009pp 185-188

[11] A. Adriansyah, and S.H.M. Amin, Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight,Regional Conference on Engineering and Science, Johor, 2006

[12] A.Nikabadi, M.Ebadzadeh , Particle swarm optimization algorithms with adaptive InertiaWeight : A survey of the state of the art and a Novelmethod, IEEE journal of evolutionary computation , 2008

[13] Gao Yue-lin, Duan Yu-hong, A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy International Conference on Computational Intelligence and Security Workshops, IEEE 2007

[14] Y. Feng, G.F. Teng, A.X.Wang, and Y.M. Yao., Chaotic InertiaWeight in Particle Swarm Optimization, In Innovative Computing, Information and Control, 2007. ICICIC07. Second International Conference on, page 475. IEEE, 2008

[15] K. Kentzoglanakis and M. Poole., Particle swarm optimization with an oscillating Inertia Weight, In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 17491750. ACM, 2009

[16] Y. Gao, X. An, and J. Liu., A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation, In Computational Intelligence and Security, 2008. CIS08. International Conference on, volume 1,IEEE, 2008, pp. 61-65

[17] G. Chen, X. Huang, J. Jia, and Z. Min., Natural exponential Inertia Weight strategy in particle swarm optimization, In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, volume 1, pages 36723675. IEEE, 2006

[18] H.R. Li and Y.L. Gao., Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation, In 2009 Second International Conference on Information and Computing Science,IEEE, 2009, pp 66-69

[19] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle swarm optimization with self-adaptive acceleration coefficients", in Proc. 1st Int. Conf.Fuzzy Syst. Knowl. Discovery, 2003, pp. 264-268.

[20] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Trans. Evol. Comput., vol. 8, no. 3, pp. 240-255, Jun. 2004.

[21] M. Senthil Arumugam, Aarthi Chandramohan, M. V. C. Rao, Competitive Approaches to PSO Algorithms via New Acceleration Coefficients Variant with Mutation Operators, Sixth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA05, PP 225-230

[22] Jing Jie, Jianchao Zeng, Particle Swarm Optimization with Diversity-controlled Acceleration Coefficients, Third International Conference onNatural Computation (ICNC 2007),0-7695-2875-9/07

[23] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator", IEEE Congress on Evolutionary Computation, vol. 3, 1999, pp. 1958-1962

[24] "Fitness-distance-ratio based particle swarm optimization" Swarm Intelligence Symposium IEEE Swarm Intelligence Symposium, 2003. SIS ’03 pp. 174-181

[25] Janson, S. and Middendorf, M., A Hierarchical Particle Swarm Optimizer for Dynamic Optimization Problems , Applications of Evolutionary Computing, Workshops2004(Conference proceedings), 2004, p. 511-522

[26] J. J. Liang, B. Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization”, Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China

[27] And Technical Report, Nanyang Technological University, Singapore

ISSN: 2321-8169
278– 283

Table III - Optimum Value obtained for benchmark functions with 2 dimensions

Function	Measure	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5	Variant 6
1	Min	**-1400**	**-1400**	**-1400**	**-1400**	**-1400**	**-1400**
	Max	-1400	-1400	-1400	-1400	-1400	-1400
	Average	-1400	-1400	-1400	-1400	-1400	-1400
2	Min	-1298.93	-1280.48	**-1299.7**	-1296.19	-1293.8	-1298.15
	Max	-579.301	-652.448	-956.586	2211.63	895.6392	-1231.89
	Average	-1023.96	-1103.48	-1224.71	-45.5429	-611.724	-1262.81
3	Min	**-1200**	**-1200**	**-1200**	**-1200**	**-1200**	**-1200**
	Max	-1200	-1200	-1200	-1200	-1200	-1200
	Average	-1200	-1200	-1200	-1200	-1200	-1200
4	Min	-1099.5	**-1100**	**-1100**	**-1100**	**-1100**	**-1100**
	Max	370.5274	843.4976	93.51586	151.1431	3586.653	2918.936
	Average	-769.789	-648.46	-608.964	-608.901	197.8021	-5.59349
5	Min	**-1000**	**-1000**	**-1000**	**-1000**	**-1000**	**-1000**
	Max	-1000	-1000	-1000	-1000	-1000	-1000
	Average	-1000	-1000	-1000	-1000	-1000	-1000
6	Min	**-900**	**-900**	**-900**	**-900**	**-900**	**-900**
	Max	-900	-900	-900	-900	-900	-900
	Average	-900	-900	-900	-900	-900	-900
7	Min	**-800**	**-800**	**-800**	**-800**	**-800**	**-800**
	Max	-800	-800	-800	-800	-800	-800
	Average	-800	-800	-800	-800	-800	-800
8	Min	**-700**	**-700**	**-700**	**-700**	**-700**	**-700**
	Max	-700	-700	-700	-700	-700	-700
	Average	-700	-700	-700	-700	-700	-700
9	Min	**-600**	**-600**	**-600**	**-600**	**-600**	**-600**
	Max	-600	-600	-600	-600	-600	-600
	Average	-600	-600	-600	-600	-600	-600
10	Min	-499.993	-499.993	**-500**	-499.973	-499.99	-499.997
	Max	-499.911	-499.973	-499.972	-499.911	-499.911	-499.96
	Average	-499.957	-499.987	-499.987	-499.942	-499.957	-499.985
11	Min	**700**	**700**	**700**	**700**	**700**	**700**
	Max	700	700	700	700	700	700
	Average	700	700	700	700	700	700
12	Min	881.3121	**800**	**800**	864.7341	862.6428	862.6428
	Max	885.444	881.3121	862.6428	938.2117	881.3121	870.1923
	Average	883.7912	841.3196	830.0384	888.6639	868.4137	865.6626
13	Min	1000	**900**	**900**	1000	**900**	**900**
	Max	1064.027	1000	1005.32	1064.027	1000	1005.32
	Average	1022.142	923.4408	941.0653	1042.637	948.2101	921.2662
14	Min	1100	**1000**	**1000**	1102.253	**1000**	**1000**
	Max	1104.419	1000	1000	1104.419	1000	1000
	Average	1101.768	1000	1000	1103.638	1000	1000

 ISSN: 2321-8169
278– 283

Table IV - Optimum Value obtained for benchmark functions with 10 dimensions

Function	Measure	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5	Variant 6
1	Min	**-1400**	**-1400**	**-1400**	**-1400**	**-1400**	**-1400**
	Max	-1400	-1400	-1400	-1400	-1400	-1400
	Average	-1400	-1400	-1400	-1400	-1400	-1400
2	Min	**-693.83**	3313.157	6705.047	3191.349	3173.716	1631.792
	Max	15245.33	24007.54	21920.99	13679.72	19234.68	18012.14
	Average	8909.491	12170.15	16524.85	9859.741	12032.87	8269.82
3	Min	-1196.89	7707.275	**-1198.89**	-843.761	11119.77	3111.577
	Max	3923968	35217157	6016.189	84154754	29194104	1527984
	Average	1336891	11487234	2382.857	32391336	12509619	425704
4	Min	2133.07	**745.7578**	3560.474	2639.793	2144.926	1698.923
	Max	5083.956	5119.249	8524.091	35908.12	6588.469	9817.251
	Average	3637.209	2710.218	6078.754	12430.18	4674.27	6341.657
5	Min	**-1000**	**-1000**	**-1000**	**-1000**	**-1000**	**-1000**
	Max	-1000	-1000	-1000	-1000	-1000	-1000
	Average	-1000	-1000	-1000	-1000	-1000	-1000
6	Min	-890.185	-890.162	-890.163	**-890.186**	-890.165	-890.151
	Max	-798.93	-829.039	-844.026	-818.523	-798.837	-798.887
	Average	-871.934	-868.708	-880.933	-875.853	-871.894	-871.886
7	Min	-733.607	-772.111	-724.68	**-784.417**	-714.763	-746.165
	Max	-657.465	-697.508	-659.811	-688.937	-651.769	-682.217
	Average	-693.78	-741.942	-693.725	-738.541	-688.183	-718.598
8	Min	-679.124	**-679.765**	-679.43	-679.15	-679.146	-679.19
	Max	-678.709	-678.859	-678.766	-678.738	-678.65	-678.639
	Average	-678.931	-679.114	-679.03	-678.882	-678.89	-678.875
9	Min	-597.177	-596.354	**-598.076**	-597.733	-595.677	-595.982
	Max	-592.403	-593.953	-593.328	-591.836	-591.644	-593.347
	Average	-594.833	-595.001	-595.364	-595.254	-594.136	-594.333
10	Min	-499.926	-499.921	-499.776	**-499.97**	-499.882	-499.899
	Max	-499.588	-499.189	-499.162	-499.714	-499.352	-499.808
	Average	-499.772	-499.593	-499.588	-499.895	-499.667	-499.862
11	Min	**1100.19**	**1100.19**	**1100.19**	**1100.19**	**1100.19**	**1100.19**
	Max	1100.19	1100.19	1100.19	1100.19	1100.19	1100.19
	Average	1100.19	1100.19	1100.19	1100.19	1100.19	1100.19
12	Min	1534.399	**1180.207**	1257.54	1438.35	1417.738	1267.159
	Max	2185.776	1782.251	1890.253	2170.613	1862.162	2054.104
	Average	1820.786	1613.171	1722.775	1882.529	1644.332	1627.448
13	Min	1698.484	1972.661	**1319.745**	1861.848	1909.81	1949.69
	Max	2447.417	2126.263	2110.018	2408.669	2440.087	2516.734
	Average	2073.518	2033.958	1859.64	2201.423	2172.421	2164.265
14	Min	**1208.801**	1213.806	1214.31	1213.304	1217.096	1220.237
	Max	1220.63	1222.862	1222.012	1223.452	1221.489	1228.262
	Average	1216.335	1220.028	1217.52	1219.165	1219.129	1223.177