



Impact of Acceleration Coefficient Strategies with Random Neighborhood Topology in Particle Swarm Optimization

Mrs. Snehal Mohan Kamalapur¹, Dr. Varsha Hemant Patil²

Research Scholar¹, Research Guide²

DYPIET¹, Vice Principal, MCERC²,

Pune, Nashik

India

Abstract: Particle Swarm Optimization is optimization technique having few parameters to tune. Inertia Weight with velocity clamping has great impact on the PSO performance. Velocity clamping is simple and effective solution to swarm explosion while inertia weight has an effect on swarm diversity. Another PSO performance enhancement parameter is the neighborhood topology. Stochastic acceleration terms c_1 and c_2 pulls each particle towards personal best and global best positions. The linearly decreasing scheme of inertia weight and velocity clamping are considered herein the paper for swarm diversity and explosion respectively. The work under consideration extends the existing parameter setting techniques on inertia weight and acceleration coefficients. The concept of a "dynamic" neighborhood is explored. The paper focuses on linear varying inertia weight, random acceleration coefficient and random neighborhood topology link based variants. Experiments have been conducted to evaluate the performance.

Keywords: acceleration coefficient; Inertia Weight component; neighborhood topology; PSO

I. Introduction

Particle swarm optimization (PSO) is optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995[1, 2]. The term PSO refers to a relatively new family of algorithms that is used to find optimal or near to optimal solutions to problems. 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. In particular, PSO incorporates swarming behaviors observed in birds and fish. PSO technique provides an evolutionary based search. PSO has very few parameters and is easy to implement so it has rapidly progressed in recent years. It has been applied to solve many real-world optimization problems. In PSO all the particles have fitness values. These values are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space. The particles fly by following the current optimum particles.

There are two main strategies for the improvements of PSO the parameter modification and population diversity. The parameter modification revises the parameters of the original PSO. Parameters include the inertia weight, the constriction coefficient, etc. Population diversity methods avoid the premature convergence caused by the loss of the population diversity. This includes different topology. Adaptive control strategy can be developed based on inertia weight [5-8] and acceleration coefficient [9-12]. Based on these improvements different variants with inertia weight, random acceleration coefficient and random neighborhood topology links are proposed here. Section II explains PSO algorithm, formulation and parameters used in PSO. Benchmark functions used for experimentation are described in section III. Section IV illustrates proposed variants. Section V focuses on results and analysis. The paper concludes in section V.

II. Basic Particle Swarm Optimization algorithm

Particle swarm optimization (PSO) is optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995[1].

A. Algorithm

- Initialize the population - locations and velocities
- Evaluate the fitness of the individual particle (local Best or pbest). Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has reached so far.
- Keep track of all individuals' highest fitness (global Best or gbest)
- Modify velocities depending on local best and global best position
- Update the particles position
- Terminate if the condition (depending on application) is met else go to step b

B. Parameters of PSO

Swarm size

It is quite a general practice in the PSO literature to limit the number of particles to the range 20 to 60 [4, 3]. It can also be computed by a formula on the dimension D.

$$S = 1 + \sqrt{D}$$

Learning Factors: lbest and gbest

In global best PSO (gbest) all the particles are neighbors of each other that is there is one neighborhood. Thus, the position of the best overall particle in the swarm is taken into consideration as the social term. It is assumed that with gbest learning swarms converge fast. However the swarm can be trapped in local optima.

In Local best PSO (lbest) only a K number of particles are neighbors. The swarm may converge slower but probability of locating the global optimum is more.

Neighborhood Topology

It is a scheme for determining the neighbors of each particle. Ring and Star are two common neighborhood topologies. In ring particle is connected only to its immediate neighbor. The gbest is special case of ring, called as star topology.

V_{max}

The maximum velocity is represented as: V_{max}

The velocity $v_{i,d}(t)$ of each particle is clamped to a maximum velocity V_{max} . Larger value of V_{max} facilitates global exploration, while a smaller value of V_{max} encourages local exploitation. V_{max} simply disabled the particle's oscillations from exploration to exploitation [4].

$c1$ and $c2$

$c1$ and $c2$ are acceleration constants.

Initial version of PSO used values, $c1 = c2 = 2$. The experimental results indicate that $c1 = c2 = 0.5$ might provide even better results []. It is better to choose a larger cognitive parameter, $c1$, than a social parameter, $c2$, but with $c1 + c2 \leq 4$ (Carlisle and Dozier, 2001).

Random Vectors : $r1$ and $r2$

The parameters $r1$ and $r2$ are used to maintain the diversity of the population. They are uniformly distributed in the range [0, 1].

Inertia Weight

In 1998, the concept of an inertia weight was developed to better control exploration and exploitation. The motivation was to be able to eliminate the need for V_{max} . The use of the inertia weight ω has provided improved performance. As initially developed, ω often is decreased linearly from about 0.9 to 0.4 during a run. Suitable selection of the inertia weight provides a balance between global and local exploration and exploitation.

After experimenting on the inertia weight, it was found that the maximum velocity factor that is V_{max} couldn't always be eliminated [16].

C. PSO Formulation

The PSO has multidimensional search space and attempts to find out an optimal solution for the function f . The terminology used for analysis is as follows-

For a D-dimensional search space, the position of the i^{th} particle is represented as:

$$X_i(t) = (x_{i,1}(t), x_{i,2}(t), \dots, x_{i,d}(t)) \quad (1)$$

The velocity of each particle is represented as:

$$V_i(t) = (v_{i,1}(t), v_{i,2}(t), \dots, v_{i,d}(t)) \quad (2)$$

Each particle maintains a memory of its previous best position which is represented as:

$$P_i(t) = (p_{i,1}(t), p_{i,2}(t), \dots, p_{i,d}(t)) \quad (3)$$

The best one among all the particles in the population is represented as:

$$p_{g,d}(t) = \min f(p_{i,d}(t^*)) \text{ for } \forall t^* > (t - 1) \quad (4)$$

At each iteration a new velocity value for each particle is evaluated according to its current velocity, personal best and the distance from the global best or local best position.

The new velocity value calculated using (1) without inertia weight or constriction coefficient.

$$v_{i,d}(t+1) = (v_{i,d}(t) + c_1 * r_1 * (p_{i,d}(t) - x_{i,d}(t)) + c_2 * r_2 * (p_{g,d}(t) - x_{i,d}(t))) \quad (5)$$

It is then used to calculate the next position of the particle in the search space

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad \text{for } i = 1, 2, 3, \dots, N \text{ and } d = 1, 2, \dots, M \quad (6)$$

Another important variant of standard PSO is the constriction coefficient approach of Contemporary PSO (CPSO), which was proposed by Clerc and Kennedy [14,15]. The velocity of CPSO is updated by (7)

The new velocity value is calculated using (7) with constriction coefficient

$$v_{i,d}(t+1) = \chi * (v_{i,d}(t) + c_1 * r_1 * (p_{i,d}(t) - x_{i,d}(t)) + c_2 * r_2 * (p_{g,d}(t) - x_{i,d}(t))) \quad (7)$$

The first part of (1) represents particle's previous velocity, which serves as a memory of the previous flight direction. This memory term can be visualized as a momentum, which prevents the particle from drastically changing its direction and biases it towards the current direction. The second part of (1) is called the cognition part. This is the personal experience of the particle. The cognition part resembles individual memory of the position that was best for the particle. The third part represents the social component. Each particle is also drawn towards the best position found by its neighbor.

III. Benchmark Functions

As we wish to test the adaptability on different type of test functions we chose one unimodal function and 2 multimodal benchmark functions [17]. All functions are tested on two and ten dimensions.

According to their properties, these functions are divided into two groups:

- A. Unimodal problems
- B. Multimodal problems,

A. Group A: Unimodal problem

Sphere function

Sphere is unimodal function. It is easy to solve. Function has the following definition

$$f(x) = \sum_{i=1}^n x_i^2 \quad (8)$$

Search area is usually restricted to $-5.12 \leq x_i \leq 5.12$, $i = 1, \dots, n$. Global minimum $f(x) = 0$ is obtainable for $x_i = 0$, $i = 1; \dots; n$.

B. Group B : Multimodal problems

1) Rosenbrock's function

Rosenbrock function is treated as multimodal function. Rosenbrock function with dimension = 2 is also known as banana function. The global optimum lays inside a long, narrow, parabolic shaped flat valley. Convergence to the global optimum is difficult to find so this problem has been often used to test the performance of optimization algorithms.

Function has the following definition

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2] \quad (9)$$

Search area is usually restricted to hypercube $-2.048 \leq x_i \leq 2.048$, $i = 1, \dots, n$. Its global minimum equal $f(x) = 0$ is obtainable for x_i , $i = 1; \dots; n$.

2) Rastrigin's function

Rastrigin's function is highly multimodal. However, the location of the minima are regularly distributed.

Function has the following definition

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

Test area is usually restricted to hypercube $-5.12 \leq x_i \leq 5.12$, $i = 1, 2, \dots, n$. Its global minimum equal $f(x) = 0$ is obtainable for $x_i = 0$, $i = 1, 2, \dots, n$.

Table I and II lists out the parameter settings of benchmark functions and PSO respectively.

Table I Parameter Settings of Benchmark Functions

Function	Dimension	Error
Sphere	Two	1e-10
	Ten	
Rastrigin	Two	1e-10
	Ten	
Rosenbrock	Two	1e-10
	Ten	

Table III Parameter Settings of PSO

Parameter	Value
Swarm Size	20 or 40 depending on size of dimension
c1 and c2	Random value between 0 to 1.4 with uniform distribution
Inertia Weight	Linearly decreasing from 0.9 to 0.4
Neighborhood Topology	Adaptive Random

IV. Proposed Variants

Adaptive neighborhood topology has been defined in [3]. It is particular case of "stochastic star"[13]. Each particle informs random K particles and also informs itself. The random value is generated at the beginning. The graph of information links among the particles is thus set at the beginning. The parameter value K is set to 3. After each unsuccessful iteration the graph of information link is modified. Thus neighbourhood of a particle changes over time. Different variants experimented here are listed in table IV

Table IV Parameter Settings of PSO

Type of Variant	Neighbourhood topology	c1 and c2
Variant 1	At the beginning and after each unsuccessful iteration the graph of information link is modified	Random value between 0 to 1.4 with uniform distribution in every iteration using uniform distribution.
Variant 2	At the beginning and after each unsuccessful iteration the graph of information link is modified	Random value between 0 to 1.4 with uniform distribution after unsuccessful iteration using uniform distribution.
Variant 3	At the beginning the graph of information link is setup	Random value between 0 to 1.4 with uniform distribution in every iteration using uniform distribution.
Variant 4	At the beginning the graph of information link is setup	Random value between 0 to 1.4 with uniform distribution after unsuccessful iteration using uniform distribution.

The experiments are carried out based on the setup as explained in section III. In the experiments the swarm size is set as 20 if number of dimensions are less than 4 and swarm size is set as 40 if dimensions exceeds 4 for all benchmark functions. Inertia weight is linearly decreasing as shown in eq(11)

$$\omega(\text{iter}) = \omega_{\max} - ((\omega_{\max} - \omega_{\min}) / \text{maximum no. of iterations}) * \text{iteration} \quad (11)$$

Maximum numbers of 15000 iterations are used. The termination criteria is either maximum number of iterations or error allowed for function value..

V. Experimental Results

To verify the proposed variants, the benchmark functions discussed in section III are used. For each benchmark function and two and ten dimension 30 runs are taken. Sample variations of calculated values for function of benchmark test with dimension 10 are shown Fig. 1 to fig.

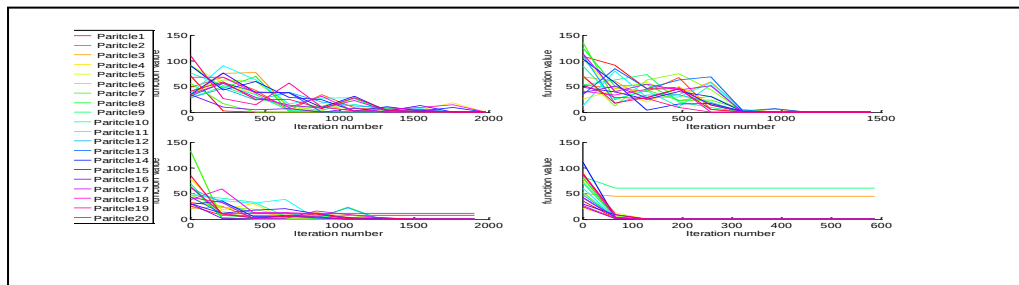
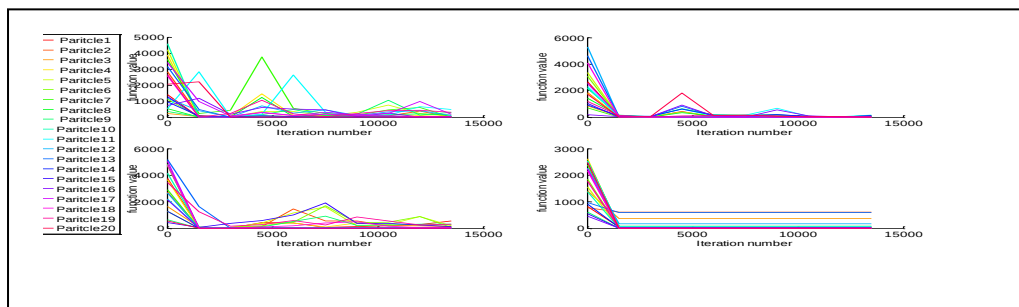
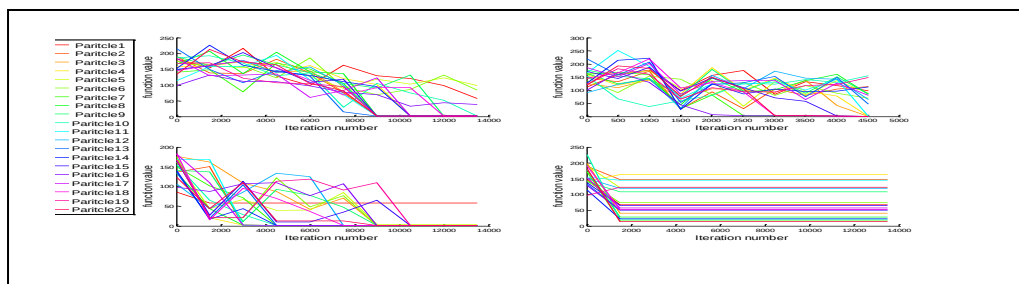
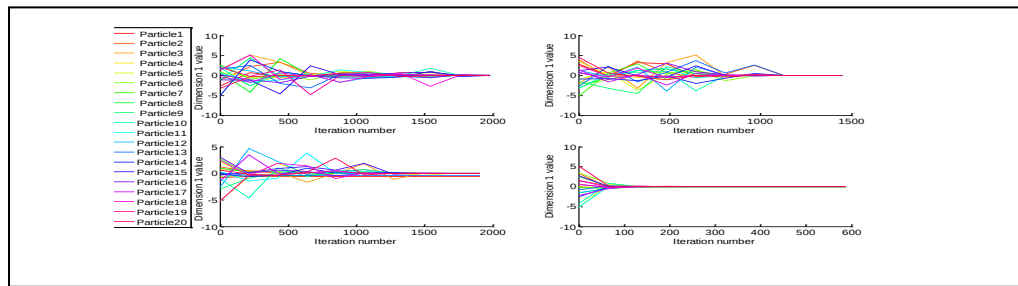
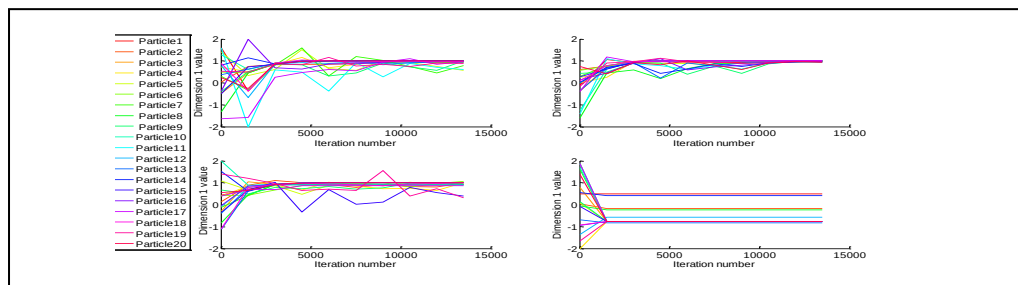
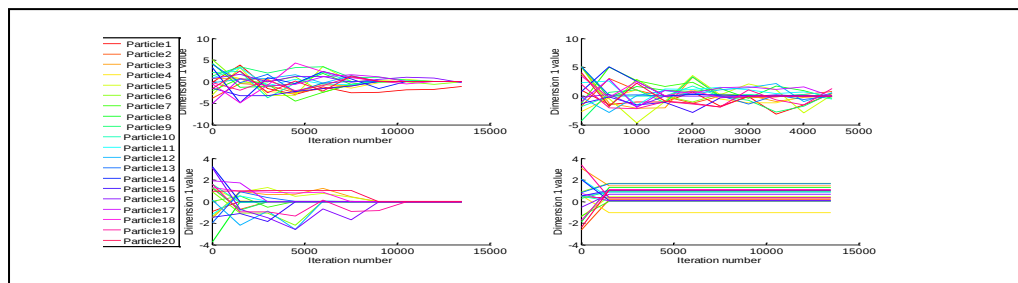
Figure 1 Sphere D10 Function value variation during execution**Figure 2 Rosenbrock D10 Function value variation during execution****Figure 3 Rastrigin D10 Function value variation during execution**

Figure 4 SphereD10 Dimension 1 of input value variation during execution**Figure 5 RosenbrockD10 Dimension 1 of input value variation during execution****Figure 6 Rastrigin D10 Dimension 1 of input value variation during execution**

The Table IV shows average number of iterations required for each benchmark functions depending on termination criteria as either error $\leq 1e-10$ or number of iterations are equal to maximum number of iterations that is 15000.

Table III Parameter Settings of PSO

Average Number of Iterations / Type of Variant	SphereD2	SphereD10	RosenbrockD2	RosenbrockD10	RastriginD2	RastriginD10
Variant 1	521.333	1868.14	1463.3333	15000	723.77	15000
Variant 2	531	1511	1403.6666	15000	760.33	11559.4
Variant 3	564.666	1719	1282.3333	15000	747.8889	13050.2
Variant 4	157.8333	6088	5171.5	15000	5146	15000

VI. Conclusion

The parameter modification and population diversity methods are used for the improvements of PSO. Unimodal and multimodal benchmark functions are used for experimentation. Acceleration coefficients is modified at each iteration or after unsuccessful iteration as parameter modification strategy. The neighborhood topology is randomly defined and modified only at the beginning or after each unsuccessful iteration. The variant using the

graph of information link modification at the beginning and after each unsuccessful iteration with random c_1 and c_2 values at each iteration exhibits better performance on benchmark set.

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] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.
- [5] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.
- [6] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 1945–1950.
- [7] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2001, vol. 1, pp. 101–106.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] T. Yamaguchi and K. Yasuda, "Adaptive particle swarm optimization: Self-coordinating mechanism with updating information," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Taipei, Taiwan, Oct. 2006, pp. 2303–2308.
- [12] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Adaptive multi-objective particle swarm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 2281–2288.
- [13] Vladimiro Miranda, Hrvoje Keko, and Alvaro Jaramillo Duque. Stochastic Star Communication Topology in Evolutionary Particle Swarms (EPSO). In: *International Journal of Computational Intelligent Research*, 2008.
- [14] E. Ozcan, and C. K. Mohan "Surfing the Waves"
- [15] Clerc M, Kennedy J. The particle swarm explosion, stability and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 2002;6(1):58–73.
- [16] R. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," *Proc. IEEE Int. Conf. on Evolutionary Computation*, pp. 81–86, 2001
- [17] Angeline, P. J.: *Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences*. In: Porto, V. W., Saravanan, N., Waagen, D., Eiben, A. E. (eds.): *Evolutionary Programming VII. Lecture Notes in Computer Science*, Vol. 1447. Springer, Berlin (1998) 601–610.