

Empirical Analysis of Effect of Particle Swarm Optimization Inertia Weight Strategies over Particle Swarm Optimization with Aging Leader and Challengers

Anu Sharma and Mandeep Kaur

Computer Science, Guru Nanak Dev University, Amritsar, Punjab, India

Abstract

Particle swarm optimization is the optimization technique motivated by swarm intelligence and aims to find the best solution in the swarm. Aging leader and challengers with Particle swarm optimization (ALC-PSO) is a population based optimization method which introduced the concept of aging and challenger generation in the PSO technique. This variant of PSO has been successful in preventing premature convergence of PSO and maintaining swarm diversity. In this paper, we briefly reviewed the inertia weight parameter and its strategies in PSO and experimentally analyzed the effect of inertia weight strategies on ALC-PSO performance. Comparison is drawn between PSO and ALC-PSO based on these strategies. Results are obtained using five different benchmark functions.

Keywords: Aging, leader, particle swarm optimization, convergence, population

1. Introduction

Particle swarm optimization with aging leader and challengers is a population based heuristic optimization technique developed Wei-Neng Chen, Jun Zhang, Ying Lin, Ni Chen and Zhi-Hui Zhan [1] which mimics social behavior like movement of organisms in a flock of bird or fish school. It is based on the concept of social influence and social learning. PSO utilizes a population of optimal solutions to explore the search space. Information exchange takes place between the individuals called particle of population called swarm [2]. Simplicity and ease of implementation has made PSO a popular area of research. It has wide range of applications such as fuzzy networks, power control, computer graphics, distribution, sensor and communication networks etc. PSO provides best solution for the hard problems and also used to solve real valued, binary and discrete problems [3].

This paper is organized as following. Next section covers the brief description of particle swarm optimization and PSO with Aging leader and Challengers *i.e.*, ALC-PSO. This section is followed by the brief overview of the categories of inertia weight strategies and benchmark functions. This section is followed by the experimental analysis of inertia weight strategies on PSO as well as ALC-PSO. Last section list open issues in the field of ALC-PSO which gives direction for future research.

1.1. Particle Swarm Optimization

This heuristic population based method consists of a problem having a swarm of probable particles, and moving these particles around in the search-space as per velocity and position update rule. A population of particles randomly positioned in an n-dimensional search space is initialized in PSO. Every particle in the swarm maintains two vectors *i.e.* velocity vector and a position vector. During each generation, each particle adhere update rules to update its velocity and position by knowing from the particle's previous best position and the best position found by the entire swarm so far. Let v_i and x_i

be the velocity and position vector respectively and M be no of particle in the search space or swarm. The update rules in the standard PSO are defined as

$$v_i^j \leftarrow v_i^j + c_1 \cdot r_1^j \cdot (pbest_i^j - x_i^j) + c_2 \cdot r_2^j \cdot (gbest_i^j - x_i^j) \quad (1)$$

$$x_i^j \leftarrow x_i^j + v_i^j \quad (2)$$

In eqn.1, pbest is the best position of a particle whereas gbest is the best position of the whole swarm. c_1 and c_2 are the two constants to measure relative performance of pbest and gbest. r_1^j and r_2^j are random numbers distributed in $[0,1]$, and $j(1 < j < n)$ represents the j^{th} dimension of the search space.[4]

Comparison among particles is required for finding best position in the swarm. Convergence speed and global searching ability are the two dynamics for evaluating the functioning of PSO algorithms. Original PSO algorithm exhibits fast-converging behavior as gbest updates velocities and distance. But In multimodal problems, a best position confined to a local optima may take in the entire swarm leading to premature convergence .Many PSO variants developed to improve the performance of PSO achieve the preservation of swarm diversity at the cost of slow convergence. It is difficult to avoid premature convergence without worsening the speed of convergence and the simplicity of the structure of PSO.

1.2 PSO with Aging Leader Algorithm

ALC-PSO algorithm is introduced for solving problem of premature convergence in PSO and maintains fast converging features of PSO. ALC-PSO differentiates itself from the original PSO in such a way that the leader of the swarm ages within a limited lifespan. The lifespan of leader is adjusted according to the leader's leading power. When the lifespan is expired, the leader is challenged and replaced by newly generated particles with better leading power. In ALC-PSO velocity update formula, gbest is replaced by particle with the best leading power *i.e.*, leader .The velocity and position update rules for ALC-PSO are given as follows:

$$v_i^j \leftarrow w \cdot v_i^j + c_1 \cdot r_1^j \cdot (pbest_i^j - x_i^j) + c_2 \cdot r_2^j \cdot (Leader^j - x_i^j) \quad (3)$$

$$x_i^j \leftarrow x_i^j + v_i^j \quad (4)$$

w in the equation (3) is inertia weight whose large value leads to global search and smaller value leads to local search. value of inertia weight affect convergence. In ALC-PSO, as soon as the leader traps into local optima, new challengers are generated to claim leadership of swarm and lead the swarm towards best solution. On straightforward unimodal functions, it is normally simple for the leader to enhance the nature of the swarm and consequently the leader has solid driving force. For this situation, the leader has a longer lifespan to lead the swarm and the pursuit conduct of ALC-PSO is fundamentally the same to that of the original PSO. Subsequently, the fast converging feature of the PSO can be protected. Whereas on complex multimodal functions, once the leader confines to local optima, it neglects to enhance the nature of the swarm and gets matured rapidly. For this situation, new challengers rise to supplant the old leader and bring in differences.

In terms of search speed, ALC –PSO is the quickest algorithm of all other PSO algorithms. ALC-PSO figures out how to get results with high precision on these multimodal functions regarding function evaluations and execution time. ALC-PSO performs better than the other enhanced PSO variations on unimodal functions.

The Steps involved in ALC-PSO is given as follows:

Step 1: Initialization: The initial positions of all particles are generated randomly within the n-dimensional search space. Velocities of particles are initialized to 0. The best

particle among the swarm is chosen to be the Leader. The age of the leader is initialized to zero and the lifespan of the leader is set to an initial value 0.

Step 2: Velocity and Position Updating: Every particle follows the velocity update rule and the position update rule to adjust its velocity and position.

Step 3: Updating leader: For particle i ($i = 1, 2, \dots, M$), if the newly generated position is better than Leader then the new generated particle becomes the new Leader of the particular population. If best position found in this iteration is better than the leader then leader is updated. In this way the Leader represents the best solution generated by particles during the leader's lifetime.

Step 4: Lifespan Control: After the positions of all particles are updated, the leading power of the Leader to improve the entire swarm is evaluated. The lifespan b is adjusted by a lifespan controller. When the leader has strong leading power the controller increases its lifespan. On the other hand if leading power of leader is poor, then controller decreases the lifespan of leader.

Step 5: Generating a Challenger: A newly generated particle with better position challenges the Leader whose lifespan is finished.

Step 6: Evaluating the Challenger: The leading power of the newly generated challenger is evaluated and compared with the leading power of existing leader. If the challenger has enough leading power, it replaces the old Leader and becomes the new Leader. Otherwise, the old Leader continues to be the leader of the swarm.

Step 7: Terminal Condition Check: Check whether the number of function evaluations exceed the maximum evaluations, if yes then terminate the algorithm else go to step 2 for another round of iteration.

1.3 Inertia Weight Strategies

Different strategies have been evolving over time for predicting value of inertia weight component as inertia weight plays an important role in performance improvement of PSO. Performance Improvement in PSO includes prevention of swarm stagnation, high convergence speed and maintenance of swarm diversity. Techniques used in estimation of inertia weight in PSO are classified into four different types which has been discussed below.

1.3.1 Constant Inertia Weight

The fundamental PSO, introduced by Eberhart and Kennedy in 1995 [1] has no Inertia Weight parameter. In 1998, Shi and Eberhart [2] proposed the idea of Inertia Weight by presenting Constant Inertia Weight for whole search. They stated that a large Inertia Weight encouraged a global search while a little Inertia Weight supported a local search [7].

1.3.2 Random Inertia Weight

A random Inertia Weight method prompts the enhancement in convergence speed of PSO in right on time cycles of the algorithm. Different Tests demonstrated that estimation of Inertia Weight going from 0.9 to 0.4 gives the excellent results. Disregarding its capacity to converge to optimum, it gets into the local optima tackling question of more complex function. As it is hard to figure whether exploration or exploitation would be better in dynamic environments for a given time. So, an random estimation of w is chosen to handle with this issue [14, 8].

1.3.3 Time Varying Inertia Weight Strategies

Most of the PSO variants use time-varying inertia weight techniques in which the estimation of the inertia weight is determined considering the iteration number. These

strategies can be either linear or non-linear and increasing or decreasing. A linear decreasing inertia weight was acquainted and was indicated viable in enhancing the characteristics of the PSO. In this method, the value of w is linearly decreased from an initial value (w_{\max}) to a final value (w_{\min}) according to the following equation:

$$w(\text{iter}) = \frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max}} (w_{\max} - w_{\min}) + w_{\min}$$

where iter is the current iteration of the algorithm and iter_{\max} is the maximum number of iterations the PSO is allowed to continue. This methodology is extremely normal and the majority of the PSO calculations modify the estimation of inertia weight utilizing this updating scheme [8].

1.3.4 Adaptive Inertia Weight

The last class of inertia weight strategies examined in this paper are those that monitor the search situation and adapt the inertia weight value according to one or more feedback parameters. In one of the adaptive particle swarm optimization algorithm, different inertia weights are assigned to different particles based on the ranks of the particles

$$w_i = w_{\min} + (w_{\max} - w_{\min}) \frac{\text{Rank}_i}{\text{Total-Population}}$$

where Rank_i is the position of the i^{th} particle when the particles are ordered based on their particle best fitness, and Total population is the number of particles. The reasonable of this methodology is that the positions of the particles are balanced in a way that very fitted particles move all the more gradually contrasted with the lowly fitted ones.

1.4 Benchmark Functions

In this we are going to discuss the various Benchmark function on which we will test the performance of PSO and ALC-PSO. Benchmark functions are presented with the aim of giving an idea about the different situations that optimization algorithms have to face when coping with these kinds of problems.

- a) *Ackley test function*: It has several local optima that, for the search range $[-32, 32]$, look more like noise, although they are located at regular intervals. The Ackley function only has one global optimum located at the point 0. Function definition is mentioned below :

$$f_1(x) = 20 + e^{-20} \exp\left(-20 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right)$$

- b) *Sphere test function*: It is one of the most simple test functions available in the specialized literature. This test function can be scaled up to any number of variables. It belongs to a family of functions called quadratic functions and only has one optimum in the point 0. The search range commonly used for the Sphere function is $[-100, 100]$ for each decision variable. Function definition for spherical function is given as follows

$$f_2(x) = \sum_{i=1}^n x_i^2$$

- c) *Schwefel test function*: It is a quadratic function. It also has only one optimum and its search range is the same as that of the Sphere function (i.e., $[-100, 100]$ for each variable). Schwefel test function is given as

$$f_3(x) = -\sum_{i=1}^n x_i \sin(\sqrt{|x_i|})$$

- d) *Cigar Test Function* : This is a multimodal objective function whose search range is $[-100, 100]$ with the function definition mentioned below

$$f_4(x) = x_1^2 + 10 \sum_{i=2}^n x_i^2$$

e) *SumCan Test Function* :This is a Objective Function whose search range is [-0.16,0.16] and global optimum at -10^5 .Function definition of sumcan benchmark function is

$$f_5(x) = -(10^5 + \sum_{i=1}^n |y_i|)^{-1}$$

2. Experimental Settings and Results

Researchers depend on empirical studies to examine the behavior of an algorithm. So, in this section we examined the performance of the PSO and ALC-PSO by evaluating their gbest values when different inertia weight strategies are employed in the velocity update equations of both algorithms. Evaluation of the gbest value of particles in swarm is performed at 5 different benchmark functions and graphs are generated in which error is plotted against iterations. Error specifies the deviation of the results from the optimum value.lesser the deviation, better the result. From results, it is evident that ALC-PSO outperforms PSO on most of the test problems.

Parameter setting for the experiment

Social Constant	2
Cognitive Constant	2
Initial population	10
Maximum Iterations	20
Inertia Weight	Set according to strategies

Table 1. Comparison of PSO and ALC-PSO using Constant Weight Strategy

Algorithm \ Function	PSO	ALC PSO
Ackley Function	0.0111	0.0505
SumCan Function	88.9310	0.0384
Cigar Function	0.6855	0.00001
Schwefel Function	0.9777	0.0001
Sphere Function	0.0000	0.0975

Table 2. Comparison of PSO and ALC-PSO using Random Inertia Weight Strategy

Algorithm \ Function	PSO	ALC-PSO
Ackley Function	0.0330	0.0098
SumCan Function	100	0.0293
Cigar Function	1.1580	0.00009
Schwefel Function	1.0062	0.0040
Sphere Function	0.0520	0.0772

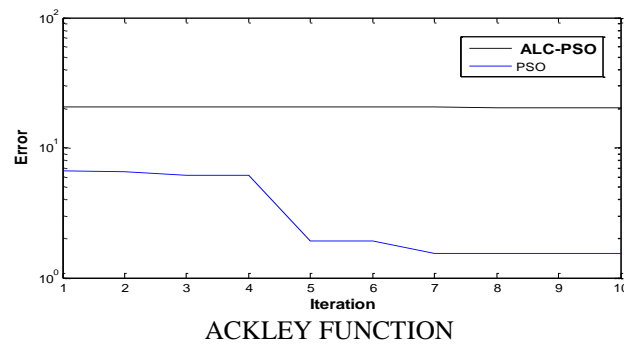
Table 3. Comparison of PSO and ALC-PSO using Chaotic Random Inertia Weight Strategy

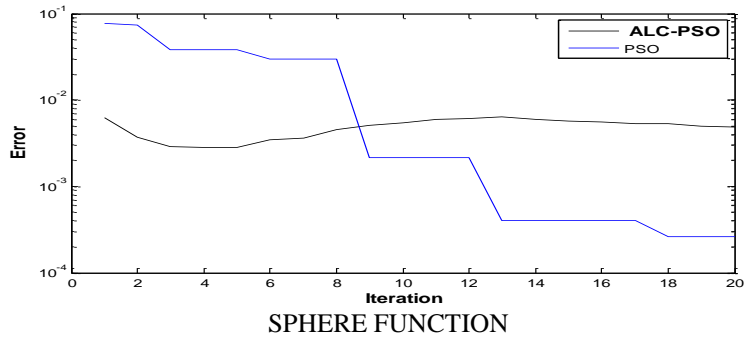
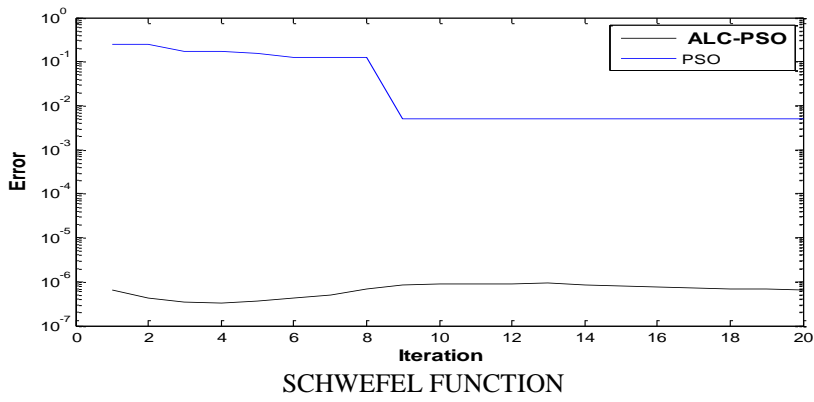
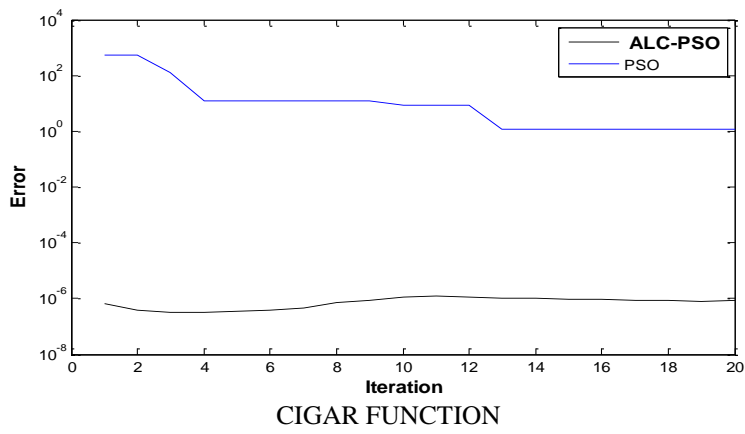
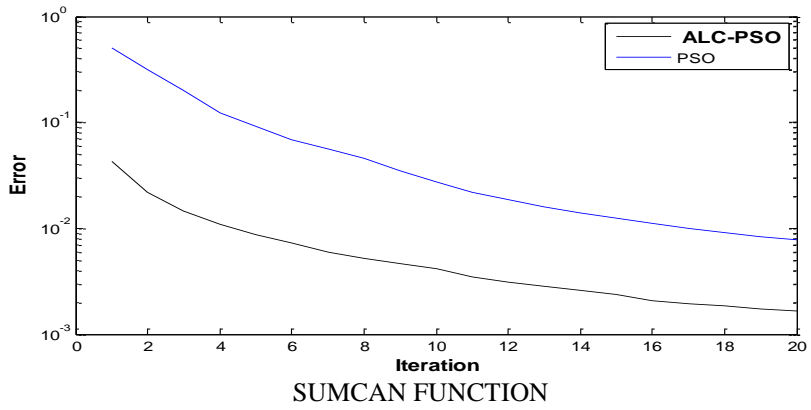
Algorithm \ Function	PSO	ALC-PSO
Ackley Function	0.0579	0.0499
SumCan Function	80.6571	0.0297
Cigar Function	0.0007	0.00001
Schwefel Function	0.9990	0.0013
Sphere Function	0.0060	0.0518

Table 4. Comparison of PSO and ALC-PSO using Adaptive Inertia Weight Strategy

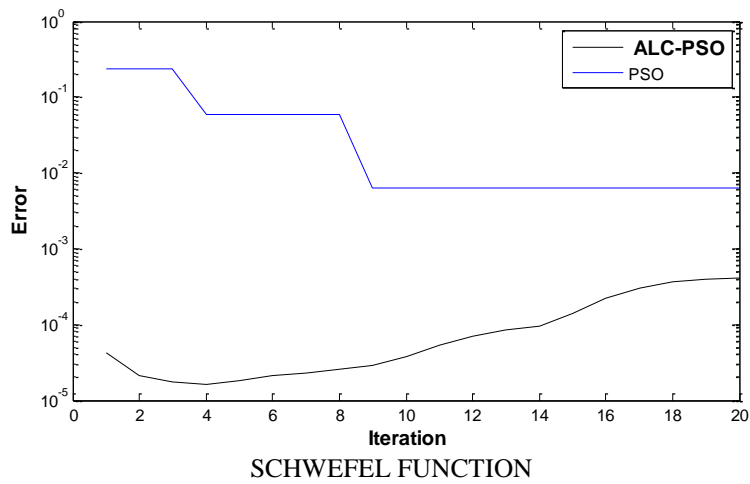
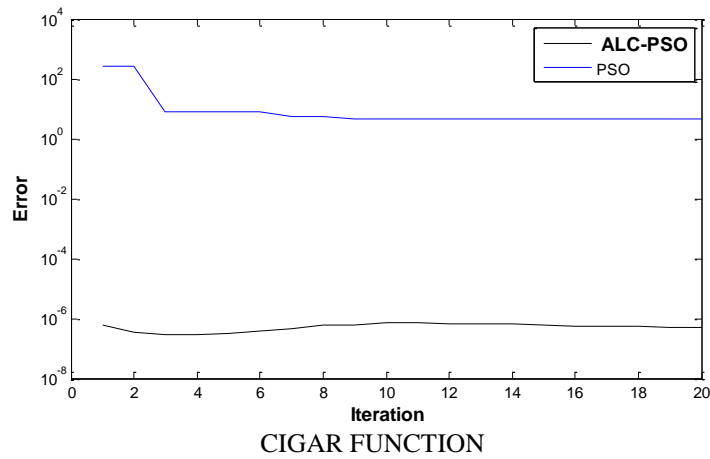
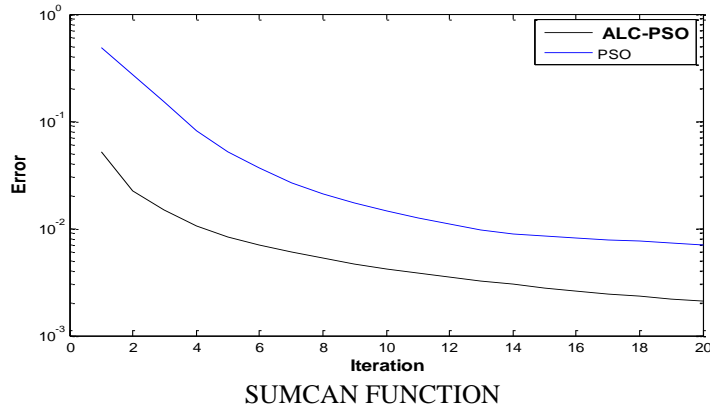
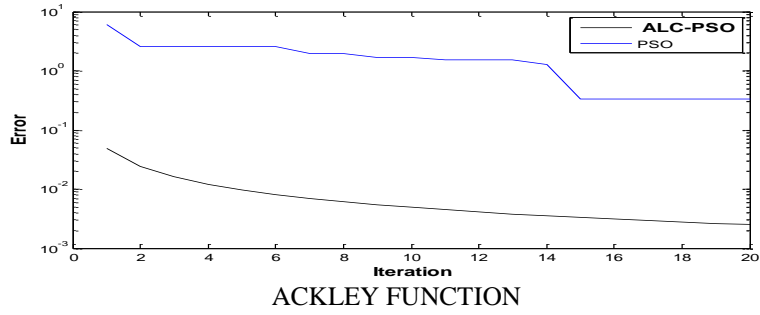
Algorithm \ Function	PSO	ALC-PSO
Ackley Function	0.0773	0.0499
SumCan Function	100	0.0405
Cigar Function	5.0337	0.00009
Schwefel Function	1.1134	0.0030
Sphere Function	0.2656	0.0342

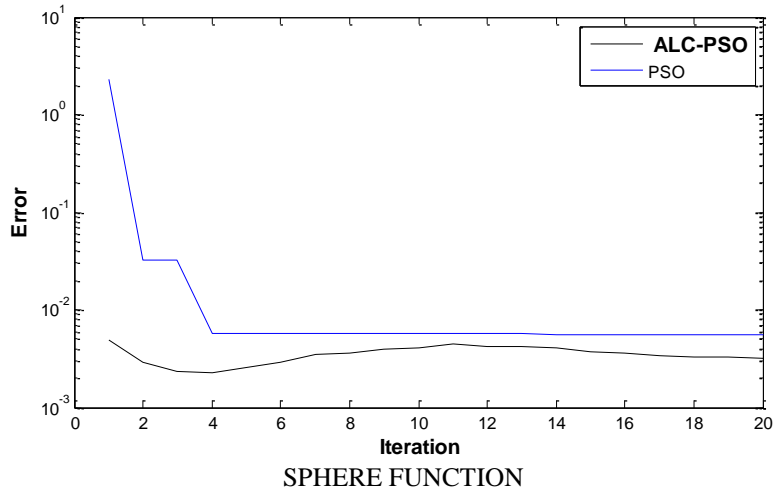
- Graphs Generated for PSO and ALC-PSO using Constant inertia weight strategy



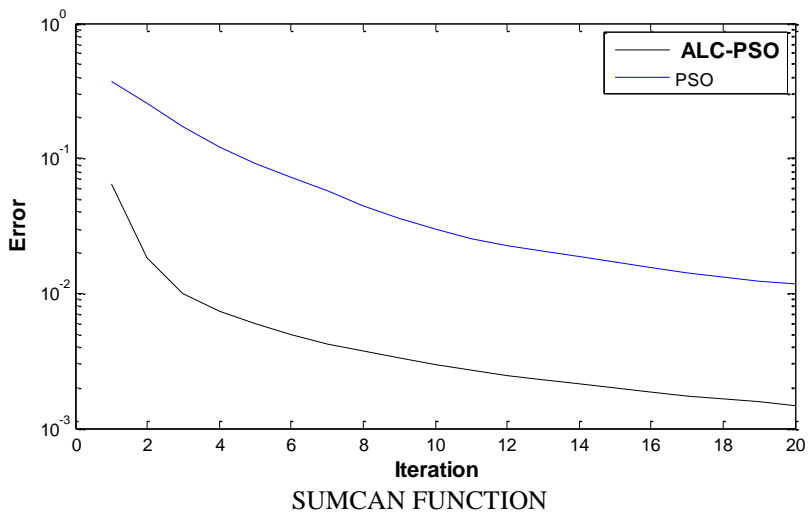
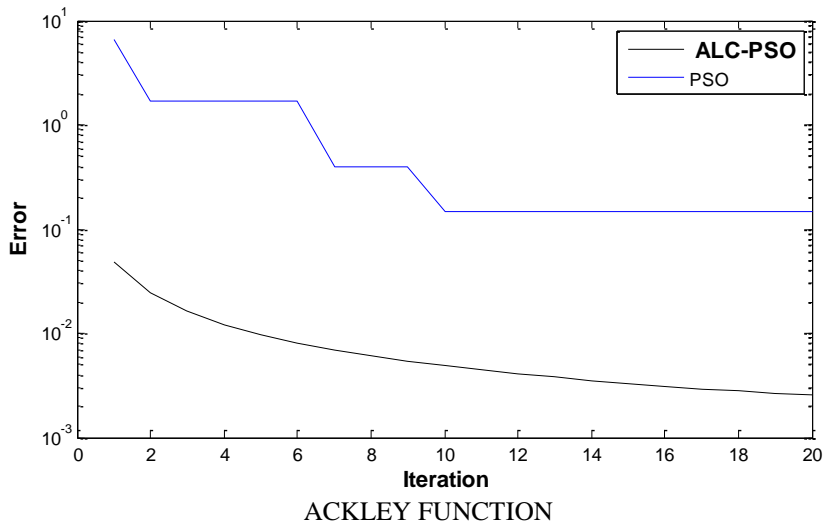


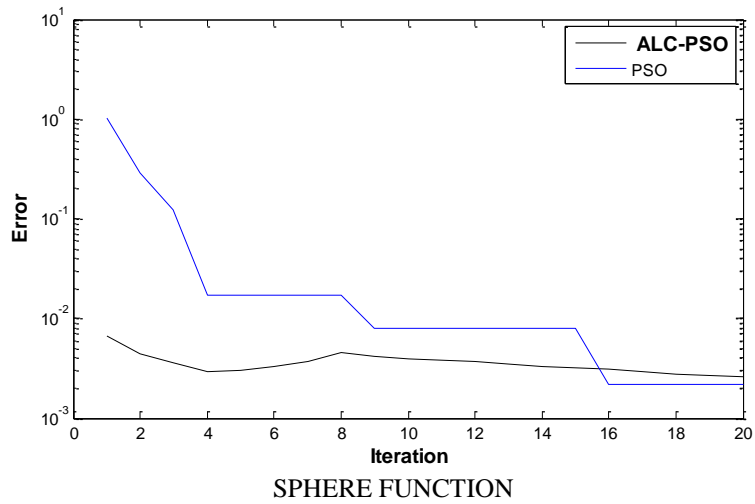
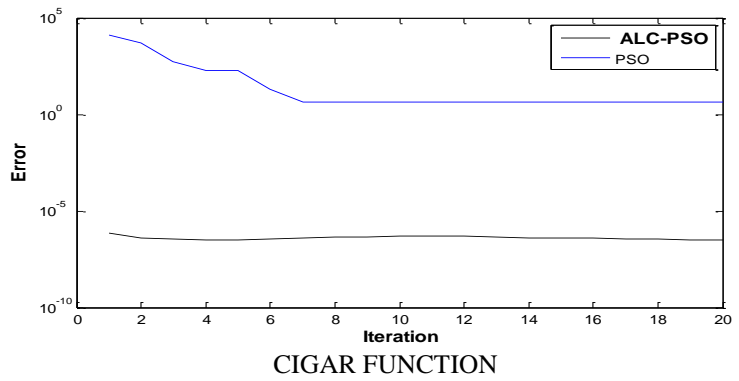
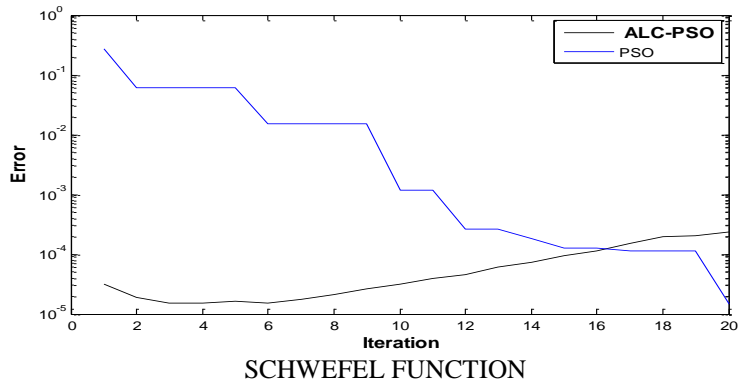
- Graphs generated for random inertia weight strategy



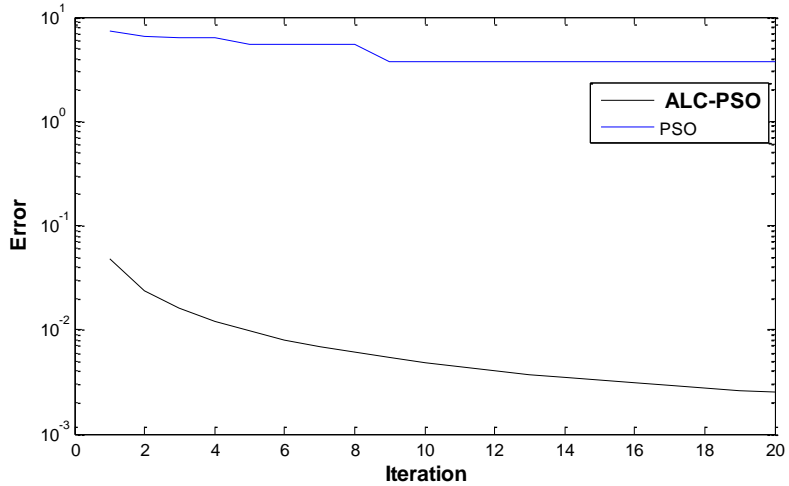


- Graphs Generated for PSO and ALC-PSO using Chaotic Random inertia weight strategy

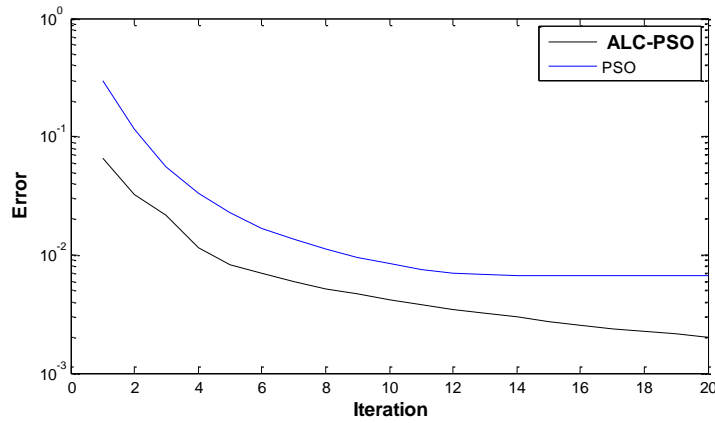




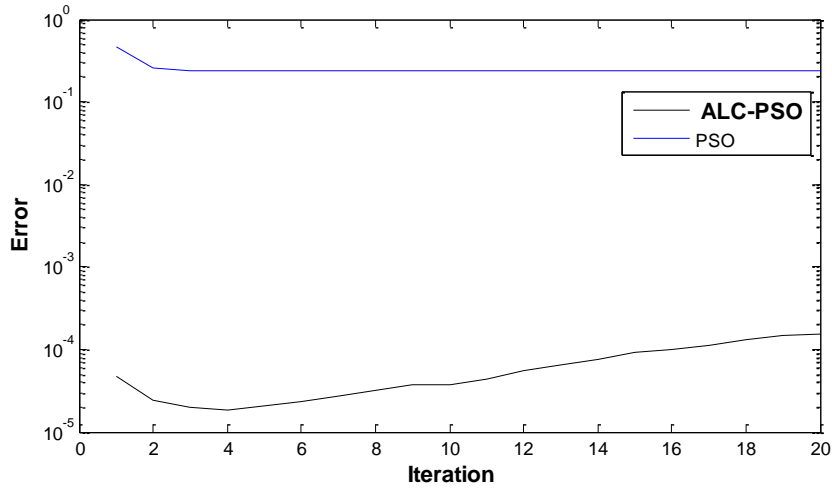
- Graphs Generated for PSO and ALC-PSO using Dynamic adaptive inertia weight strategy



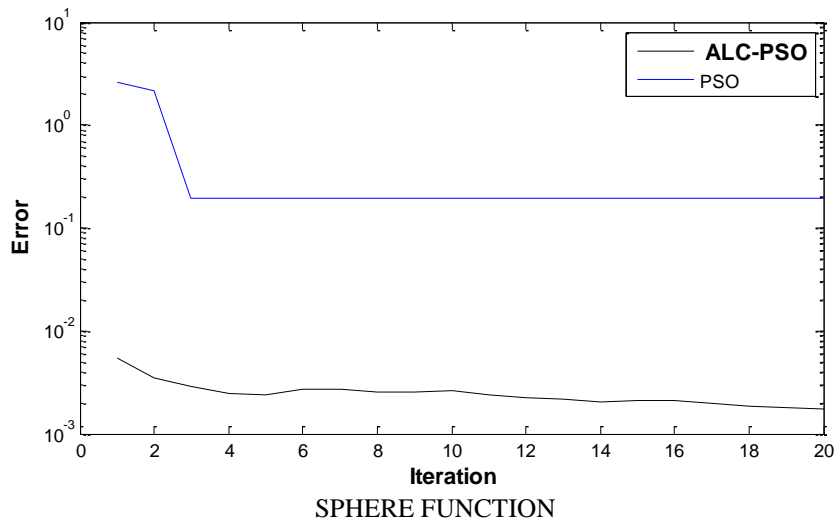
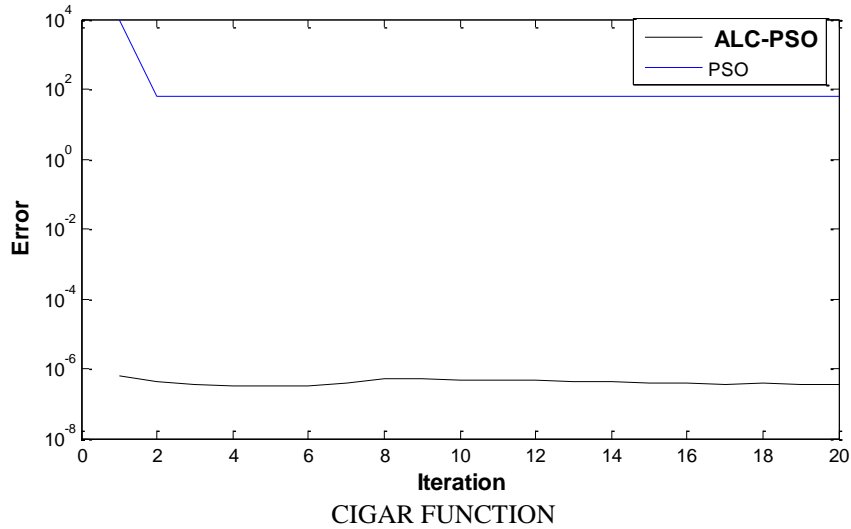
ACKLEY FUNCTION



SUMCAN FUNCTION



SCHWEFEL FUNCTION



3. Future Scope

Considerable amount of work has been done in direction of improving performance of PSO by introduce numerous variants but there is always scope of improvement so some of the possible work given below can be addressed in future to analyze the behavior of ALC-PSO algorithm .

- In future we can test the performance of ALC-PSO on other benchmark functions as well.
- Changes in update equations can be considered to evaluate its impact on algorithms performance.
- Concept of Aging can be introduced in other evolutionary computation techniques.
- Effect of the others parameter variation like constriction coefficients can be evaluated on ALC-PSO.

4. Conclusion

So far, Inertia weight strategies have been used in the particle swarm optimization algorithm. In this paper, Effect of four categories of inertia weight strategies has been evaluated in the ALC-PSO. From results, it is evident that inertia weight strategies had a

positive impact on the performance of the PSO as well as ALC-PSO for most of the test problems. But when PSO and ALC-PSO is compared on the basis of performance then ALC-PSO came out to be better than PSO.

Acknowledgment

I would like to thank my teachers, parents and my friends for all their support in this paper

References

- [1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", in Proc. IEEE Int. Conf. Neura Netw., (1995) November–December, pp. 1942–1948.
- [2] R. Poli, "Analysis of the Publications on the Applications of Particle Swarm Optimisation", Journal of Artificial Evolution and Applications, vol. 2008, Article ID 685175, pp. 10, doi:10.1155/2008/685175.
- [3] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li and Y.-H. Shi, "Particle Swarm Optimization with an Aging Leader and Challengers", Ieee transactions on evolutionary computation, vol. 17, no. 2, (2013) April.
- [4] T. C. Goldsmith, "The Evolution of Aging", Crownsville, MD: Azinet Press, (2006).
- [5] L. A. Gavrilov and N. S. Gavrilova, "Evolutionary theories of aging and longevity", *Sci. World J.*, vol. 2, (2002) February, pp. 339–356.
- [6] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li and Y.-H. Shi, "Particle Swarm Optimization with an Aging Leader and Challengers", Ieee transactions on evolutionary computation, vol. 17, no. 2, (2013) April.
- [7] S. Vijayalakshmi, D. Sudha, S. M. Sigamani and K. K. Devi, "Particle Swarm Optimization with Aging Leader and Challenges for Multiswarm Optimization", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) vol. 3, Issue 3, (2014) March.
- [8] Y. Feng, G.-F. Teng, A.-X. Wang and Y.-M. Yao, "Chaotic Inertia Weight in Particle Swarm Optimization", 0-7695-2882-1/07, IEEE.
- [9] Y. H. Shi and R. C. Eberhart, "A modified particle swarm optimizer", in: IEEE International Conference on Evolutionary Computation, Anchorage Alaska, (1998), pp. 69–73.
- [10] R. C. Eberhart and Y. H. Shi, "Tracking and optimizing dynamic systems with particle swarms", in: Congress on Evolutionary Computation, Korea, (2001).

Authors



Er. Anu Sharma, M. Tech. Computer science and engineering, from guru nanak dev university, Regional Campus, Jalandhar, Punjab (India). Has published 5 papers in conferences and journals.



Er. Mandeep Kaur, Lecturer, Computer Science department in Guru Nanak Dev University, Regional Campus, Jalandhar (India). An expert in the field of "System Simulation". Has an experience of 15 years and published fifteen papers in conferences and journals.

