International Journal of Hybrid Information Technology
Vol.8, No. 5 (2015), pp. 135-144
http://dx.doi.org/10.14257/ijhit.2015.8.5.15

Implementing Particle Swarm Optimization with Aging Leader and Challengers (ALC-PSO)

Er. Avneet Kaur[1] and Er. Mandeep Kaur[2]

[1] Student ,Computer Science Department,
Guru Nanak Dev University,
Regional Campus, Jalandhar.
[2]Lecturer, Computer Science Department ,
Guru Nanak Dev University,
Regional Campus, Jalandhar, India.
avneet.kaur.arora1990@gmail.com[1], mandeep8@gmail.com[2]

Abstract

In nature, the organisms have a limited lifespan and they grow older with time. Aging is an essential process which leads to the maintenance of species diversity in environment. Every group of species is lead by a leader. As the lifespan of every organism is limited, at a certain point of its life time, the organism deteriorates and become inefficient to lead its group. In this situation, a new leader is found who can efficiently lead its group. The lifespan of the leader and its leading power is checked, if it is not efficient enough, a new challenger is found to lead the group. This aging mechanism is applied to the stochastic process of Particle Swarm Optimization(PSO), in order to remove the limitations that existed in PSO such as: it gets stuck in local optima and the algorithm converges pre-maturely. When aging leader algorithm is applied to PSO, these limitations are removed in an efficient manner. This paper presents some issues that occur while designing and implementing a variant of PSO (Particle Swarm Optimization) i.e. ALC-PSO (PSO with Aging Leader and Challengers) which can highly improve the performance of PSO by applying the process of aging to the members of the swarm , bringing its members to the best position.

Keywords- *Aging, Benchmark functions, Best Position, Challengers, Leader, Optimization, Particle, Premature Convergence, and Swarm*

1. Introduction

PSO is a heuristic global optimization method [1]. It is based on Swarm Intelligence [2]. It is an optimization technique based on stochastic behavior of population [6]. It is an Artificial Intelligence technique, which can find approximate solution to some difficult problems. PSO is a biologically inspired optimization method [4]. PSO uses swarming behaviors observed in bird flocking, fish schooling, bee swarming and socially interactive behavior of humans. PSO is based on the bird flocking and in search of food, the birds scatter to different places. In every bird flock, there is a bird that can sense the food, all other birds follow it and accumulate at the place where food can be found [1].

Aging is a progressive process [5], which is inevitable in nature. In reality, aging is a universal process [5], which maintains the balance among species and makes the population grow at a normal pace, bringing diversity in species [3]. Organisms grow older *i.e.* they age with time. Aging is an important and intrinsic process.

There is a leader of the population who is best among the group in certain qualities. It leads the members of the population. This leader also ages with time and becomes weaker. It cannot lead the population efficiently. Then arises the need of a new leader, who can actively lead its group. This deleterious process of aging leads to challenge the position of the leader leading a group of organisms and makes the other and young organisms become new leader. The election of the leader from among various available challengers is done based on its leadership performance and lifespan.

According to the leading power of the leader, its lifespan is adjusted. If it has good leading power, it lives longer leading the swarm, and brings all the members of the swarm towards best position so found but if is not capable of leading the swarm, new challengers emerge as new leader , claiming the leading position in swarm.

Whenever the leader of group becomes aged, new challengers come up to lead the group. The new challengers are generated using two parameters *i.e.* performance and lifespan. The lifespan of the leader is tuned by the lifespan controller according to its leading power and new challengers are generated. Using some function evaluations, the generator continues generating the challengers till the maximum evaluations are reached. The best challenger becomes the new leader of the swarm [3].

2. Elements of Aging Leader and Challengers (ALC)

1. Leader- The leader of the group which leads whole of the group and takes them all to the best solution in the swarm. The leader must have some distinct qualities, which make it stand best in the group.

2. Challenger- Challenger is the member of the group which comes up to challenge the current leader of the group when the leader becomes weak or incapable of leading its group and inefficient in taking the group towards best position.

3. **Lifespan-** every organism lives for a limited lifespan. The age of organism can be said to be its lifespan. It is adjusted according to the leading power of the leader.

4. **Leading power-** The performance of the leader is its ability to lead the group. It is denoted by the leading power of the leader.

5. **Population**- The group of similar members is called the population. The size of population can be set according to the convenience.

3. Parameters used in ALC-PSO

1. **Iterations**- Number of times the process of generating a challenger runs is called the iteration.
2. **Lbest-** the local best position found by the algorithm
3. **Pbest-** another best position in the neighborhood of the particle
4. **Gbest-** globally best position in the whole swarm
5. **Function evaluation-** the number of times a challenger is generated.
6. **Maximum evaluations-** maximum number of times a challenger can be generated. It is fixed during the implementation of the algorithm, say (20).
7. **Number of particles-** the number of particles considered for the algorithm.
8. **Dimensions-** The algorithm is implemented in two dimensions (i,j).
9. **Boundary values-** The upper and lower bounds for the positions of the particles is set.
10. **Location-** The location of the challenger is found in the multi-dimensional space.
11. **Velocity-** The velocity of the particle is initialized and updated. v_{max}- maximum set velocity of the particle, v_{min}- minimum set velocity of the particle
12. **Position-** The position of the particle is initialized and updated. Xmax-maximum set position of the particle, x_{min}- minimum set position of the particle.

4. Issues in Implementing ALC-PSO

ALC-PSO algorithm can be applied efficiently on the various tools available- MATLAB, C++, JAVA, ASP.Net and many others. The issues while implementing the algorithm can be summarized as:

1. Decrease search time for selecting a leader.
2. Multi-objective and dynamic optimization requires more diversity.
3. Effects of aging on optimization problems.
4. Finding multiple globally optimal solutions in search space.
5. Combining aging mechanism with multi- population as these problems require multiple solution simultaneously.
6. Applying aging mechanism to other complex optimization problems like: GA, ACO.
7. Developing a robust algorithm.
8. The Efficiency should be more than PSO.

5. PSO with Aging Leader and Challengers—ALC-PSO

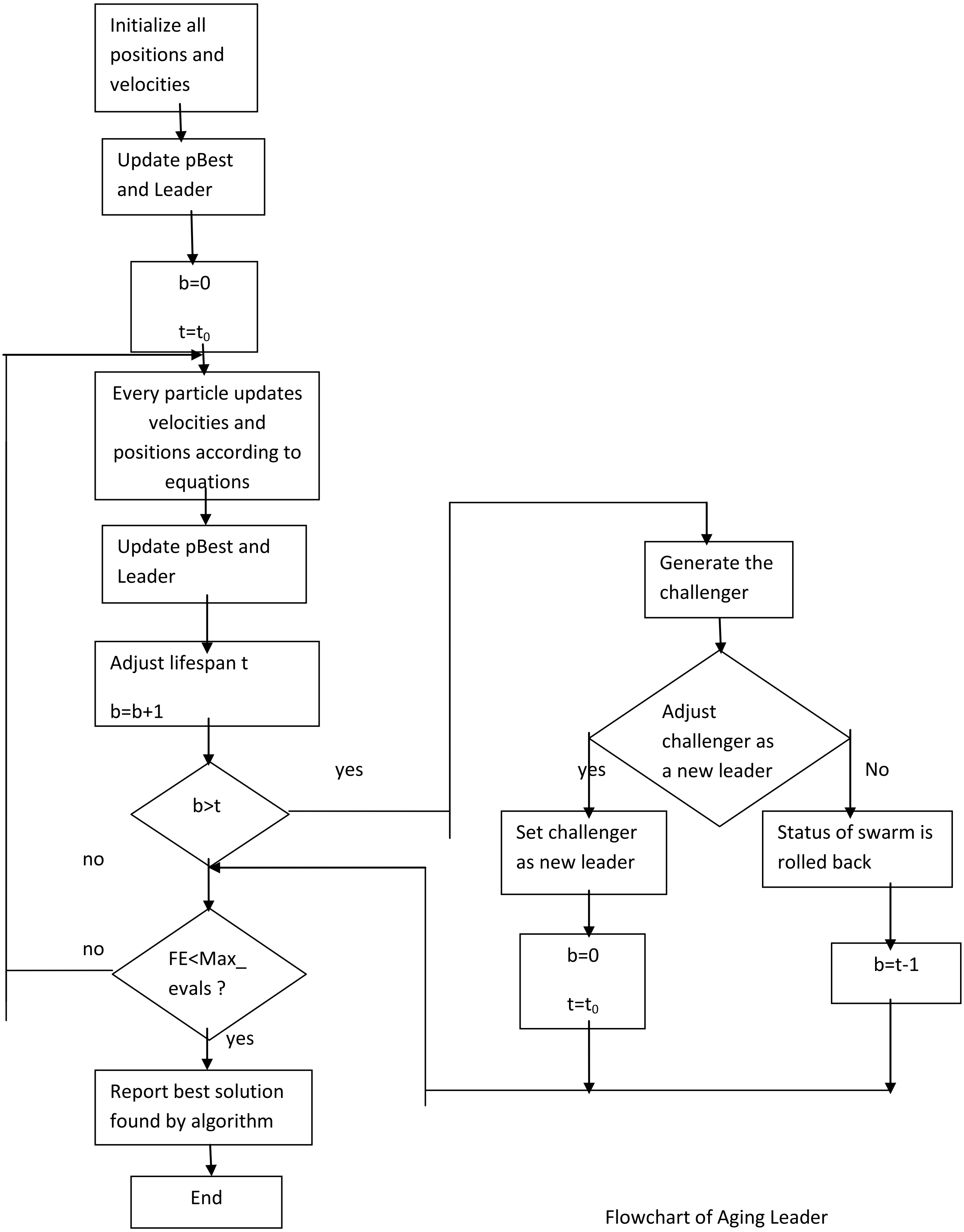
ALC-PSO is different from original PSO as in simple PSO there is no limit on the lifespan of the leader of the swarm but in ALC-PSO, the leader ages within a limited lifespan. This lifespan is dependent on the leading power of the leader of the swarm and it can be adjusted accordingly. When the lifespan of leader gets exhausted, the leader is required to be replaced by some new particle, which challenges the position of the leader and makes itself available for that position. The velocity update rule is changed to:

$_{Vi}{}^{j} = w.*_{Vi}{}^{j} + c_1.r_1{}^{j.}\,(pBest_i^j - x_i^j\,) + c_{2.}r_2{}^{j}.\ (Leader^j - x_i^j)$

Here leader is a particle with adequate leading power generated by aging mechanism.

5.1. Working Considerations for the Algorithm:

The parameters used: Age of leader- b
Lifespan of leader- t
c1, c2- parameters to weigh relative importance of $pBest_{i,,}gBest_i$
r1, r2- random numbers uniformly distributed in [0,1].
j- dimension of search space.
V_i*-* $(v_i^1, v^2{}_j,\ \dots\ v^i{}_n)$ *–*i^{th} *particle's velocity vector*
X_i*-* $(x_i^{1,}\, x^i{}_2, \dots x^i{}_n)$*-* i^{th} *particle's position vector.*
Leader- particle generated by aging mechanism with adequate leading power.
gBest $(gBest^1, gBest^2 \dots.\ gBest^n)$*- historically best position of entire swarm*
$pBest_i$ $(pBest^1, pBest^2 \dots pBest^n)$*- historically best position of particle i(i=1,2,…M)*
M- Number of particles in population


Initialize all positions and velocities
Update pBest and Leader
b=0
t=t_0
Every particle updates velocities and positions according to equations
Update pBest and Leader
Adjust lifespan t
b=b+1
b>t
yes
no
FE<Max_ evals ?
no
yes
Report best solution found by algorithm
End
Generate the challenger
Adjust challenger as a new leader
yes
No
Set challenger as new leader
Status of swarm is rolled back
b=0
t=t_0
b=t-1

Flowchart of Aging Leader

Algorith

Step ***1: Initialization***-All particles of swarm are generated randomly and initialized in the n- dimensional space.

The velocities of particles are initialized to zero.
Best particle among swarm is selected to be the leader.
Age of leader b is initialized to zero. *i.e.* b=0
Lifespan of leader *t* is initialized to its initial value *i.e.* t=t_0

Step 2: Updating velocity and position- The velocity and position of every particle in the swarm is updated according to the velocity and position rules.
$v_i^j = w.*v_i^j + c_1.r_1^{j.} (pBest_i^j - x_i^j) + c_2.r_2^j. (Leader^j - x_i^j)$
$x_i^j = x_i^j + v_i^j$
These velocities and positions are updated.

Step 3: Update pBest and Leader-If the new generated position X_i for every particle (i=1,2,….M) is better than $pBest_{i\,,}$ then X_i is made to be the new $pBest_i$If best position found in this iteration is better than the leader, then leader is updated to become the best position.

Step 4: Lifespan control-After updating the position of all particles of the swarm, the leading power of leader is evaluated. This leading power is evaluated on the basis to improve the entire swarm. Lifespan t is adjusted by lifespan controller and age of leader b is increased by 1.
If lifespan of particle gets exhausted, i.e b> t, go to step 5, otherwise go to step 7.

Step 5-Generating challenger- New particle is generated to challenge the leader , whose lifespan has got exhausted. When the lifespan becomes exhausted *i.e.* b=t, the swarm gets trapped in local optima. So, new particle is required to be generated , which can challenge the leader and tries to replace the previous leader This newly generated particle is denoted as Challenger (Challenger ^{1,} Challenger ^{2},…. Challenger ^{n})

For every dimension j(j=1,2,… n) , a random number $rnd_j$is uniformly distributed within (0,1) is generated and compared with a parameter pro € (0,1). If rnd_j< pro, challenger j is set to a randomly generated number that is uniformally distributed in range [L^j, U^j], where $L^j$and $U^j$are lower and upper bounds of dimension j, else Challenger ^{j} is inherited from the previous leader. If the newly generated challenger is exactly same as old leader, there is no significance. Iit should be different. In order to guarantee this, the dimension of challenger are derived form the leader . The algorithm randomly selects one dimension (ran) and resets $challenger^{ran}$ to a random number uniformally distributed in [L^{ran}, U^{ran}]

Step 6: **Evaluating challenger-** The leading power of newly generated challenger is evaluated, if this challenger has enough leading power, it replaces the old leader and itself becomes the new leader.
Age is initialized to 0 *i.e.* b=0
Lifespan t is initialized to $t_{0.}$ else the old leader remains unchanged and will continue leading the swarm.

Step 7: **Terminal condition check-** If number of FE (function evaluations) is larger than predefined maximum evaluation number Max_evals, the algorithm gets terminated. Otherwise, algorithm returns to step 2. And other steps are repeated [7].

6. Analysis of ALC-PSO

6.1. Analysis of Lifespan Controller

When the leader has strong leading power to lead the whole swarm, the lifespan controller increases its lifespan t but if leading power is poor, the controller reduces its lifespan t.

Evaluation of leading power is done using three terms: gBest, pBest, Leader.

Case 1: Good leading power ($D_{gbest}(b<0)$:

Leader has sufficiently good leading power and manages to lead the swarm to improve the best solution that is to be found by the algorithm. The current leader is very active and can achieve improvements. It has good potential to improve the swarm. So, lifespan controller increases lifespan t of leader by 2.

Case 2: fair leading power ($D_{gbest}(b)=0)$ and $\sum_{i=1}^{m} D_{pbest}(b)<0)$:

Though the historically best solution (pBestI) that is found by algorithm is not improved, still the leader is able to lead at least a single particle in the swarm, leading to its improvement. Leader has still some potential to improve the swarm. So, lifespan t is increased by 1.

Case 3: Poor leading power (D_{gBest} (b)=0, $\sum_{i=1}^{m} D_{pBesti}(b)=0$ and D_{leader} (b)<0):

Leader fails to lead any particle in the swarm towards better position. However, leader has still some ability to improve itself. So, the lifespan of leader remains unchanged.

Case 4: No leading power ($D_{gbest}(b)=0$, $\sum_{i=1} D_{gBesti}(b) = 0$ and $D_{leader}(b)=0$):

Neither the quality of leader nor the quality of swarm is improved. So, current leader has no potential to improve swarm, so controller reduces lifespan t by 1.

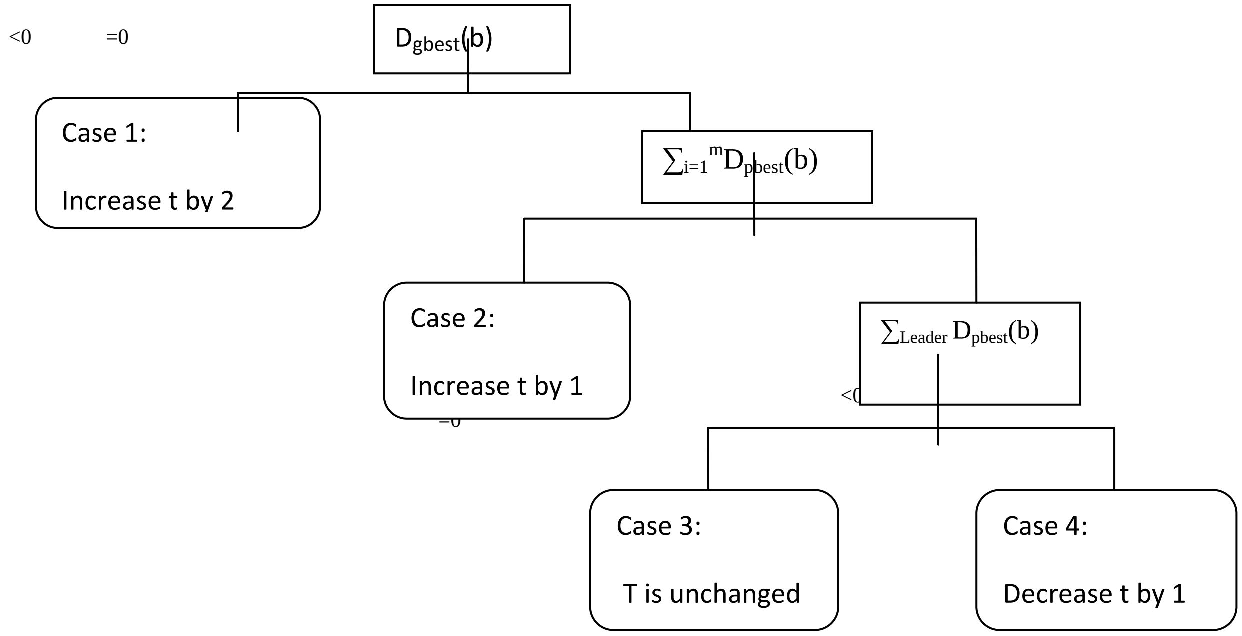


Figure. Decision Tree for the Lifespan Controller to Evaluate the Leading Power of the Leader and to Adjust the Lifespan t

6.2. Analysis of criteria for accepting a Challenger

The algorithm assumes the challenger as a temporary leader of the swarm for T iterations. Thus, velocity update rule changes to:

$$v_i^j = w.*v_i^j + c_1.r_1^{j.} (pBest_i^j - x_i^j) + c_2.r_2^j. (Challenger^j - x_i^j)$$

If any pBest I (i=1,2,… M) gets improved during the T iterations, the algorithm accepts this challenger and uses it to replace the previous leader to lead the swarm.

Age of new leader b is set to 0.
Lifespan of leader t is re-initialized to $t_{0.}$
If no best solution is found, position is improved during T iterations , previous leader is resumed.
Lifespan t of old leader remains unchanges and its age b is reset to b=t-1.
So, previous leader can still lead the swarm in next generation. If still there is no improvement then some another challenger is generated to challenge that previous leader [7].

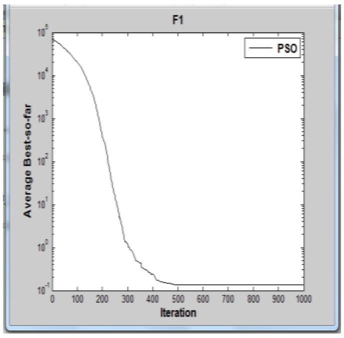
7. Experimental Settings for Implementation

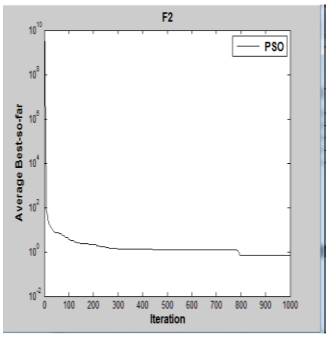
The PSO algorithm and ALC-PSO are implemented on MATLAB (R2011b). The algorithm gives the convergence point at which all the particles of the swarm get accumulated, means the optimal solution is found at that point.Results for 10 different function evaluations are calculated and tabulated.

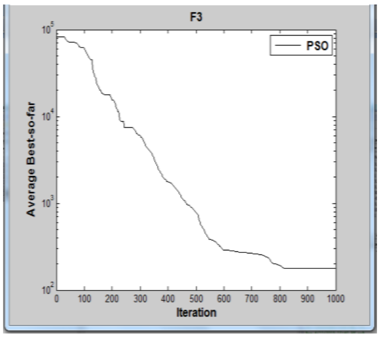
Table- Results of Function Evaluations InALC-PSO

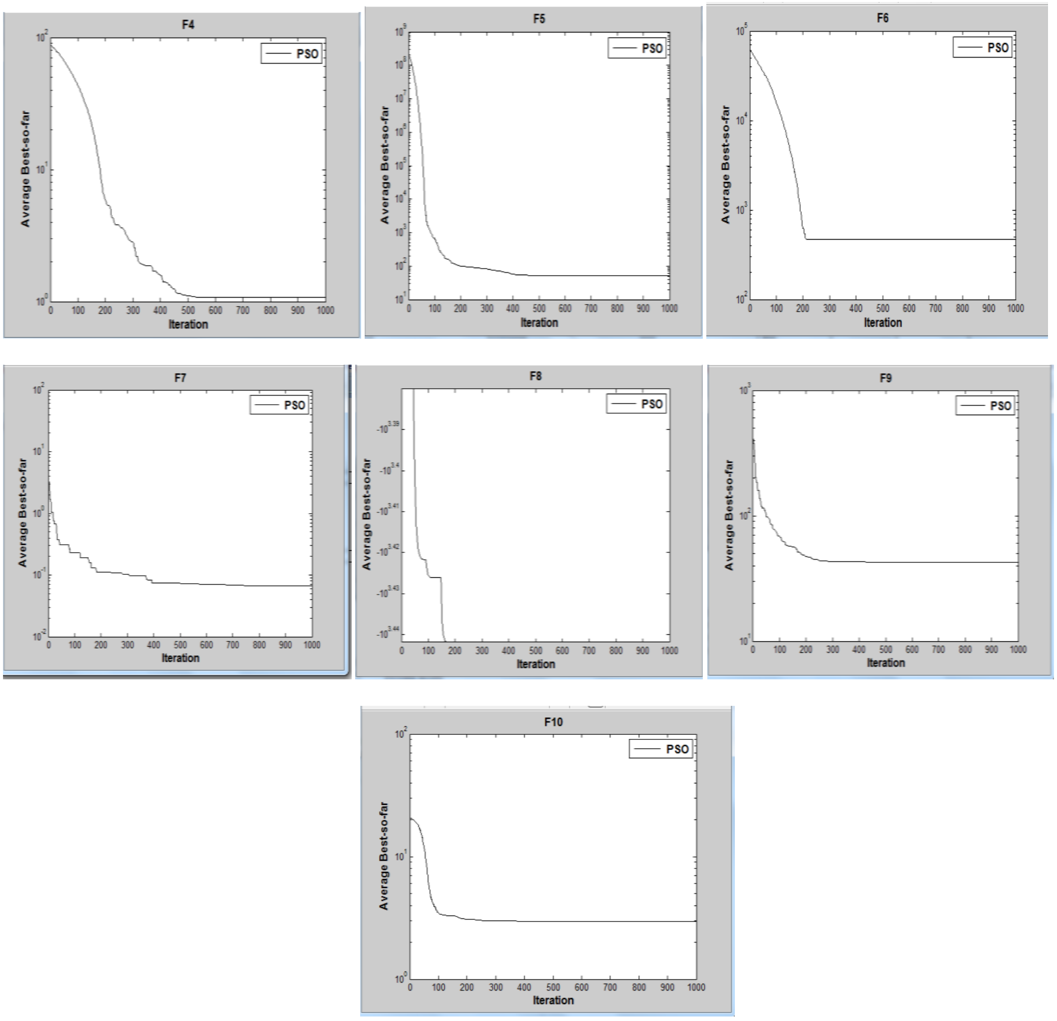
Function Evaluations	Optimum Point	MED_BST	AVG_MEAN_FITNESS
F1	0.1332	0.1332	0.1332
F2	0.7142	0.7142	0.7142
F3	177.6882	177.6882	177.6882
F4	1.0812	1.0812	1.0812
F5	51.0431	51.0431	51.0431
F6	471	471	474.4000
F7	0.0670	0.0670	0.5732
F8	-2.7660e+003	-2.7660e+003	-320.2016
F9	42.8006	42.8006	42.8006
F10	2.9713	2.9713	2.9713

MED_BST-MeanBest value, AVG_MEAN_FITNESS- Average Mean Fitness (Solution)
Function evaluations - Here, 10 function evaluations are made to check if the challenger generated is better than the leader of the swarm.
MAX_EVALS- 23









Range of Average_Best_so_far(Average best position found so far)- 10^{-2} to 10^{2}
Number of Iterations - 1000

8. Significance of ALC-PSO

1. Easy and simple algorithm.
2. Diversity is achieved.
3. Aging phenomenon and performance can be efficiently used.
4. Algorithm is efficiently defined, designed and presented.
5. It converge and escapes of local optima thus, prevents premature convergence
6. Fast converging feature of PSO is maintained.
7. Performance is enhanced as better challenger come up to lead the swarm
8. Swarm’s quality and functionality is improved.
9. It is a flexible algorithm, the lifespan of leader can be adjusted according to the leading power of leader.
10. Searching in ALC-PSO is comparable to PSO.
11. Leader can lead the swarm for long time because of large lifespan.
12. Effectively applied to many applications like: task assignment, scheduling, optical power flow etc.

9. Gaps in Study

There are some gaps in the study of ALC-PSO done above. These include:

1. Multi-objective optimization is not considered.
2. Whether aging mechanism be applied to other complex optimization problems like: ACO, genetic algorithm, is not studied.
3. Decreasing search time for selecting the leader cannot be achieved.
4. No consideration of overlapping and mutation calculation.
5. Partial optimization problem persists.

10. Conclusion and Future Scope

ALC-PSO (PSO with Aging Leader and Challengers) is a variant of PSO. Normally, PSO is applied to those behaviors, in which there is no leader to lead the group like: bird flocking and bee swarming, but in ALC-PSO, one of the members of the group is made to be the leader to lead the population and bring all of them to the best position in whole swarm. The aging mechanism is applied to the PSO, so that some parameter be set to check the performance of the leader of swarm. In case, the leader is insufficient to lead the swarm, a new leader is found which can efficiently bring the whole swarm toward a best position. The generation of challengers is done using some function evaluations. The challengers are evaluated and the best challenger is made to be the leader of the swarm, improving the best position in the swarm and thus, improving the performance of PSO algorithm. Several parameters have been studied here, which are useful in the implementation of Particle Swarm Optimization with Aging Leader and Challengers (ALC-PSO). The designing process and some of the issues related to its designing and implementation are summarized. The ALC-PSO in multi-objective optimization environment and reducing the time for the selection of leader is still a challenge, which can be a good future scope. Aging leader and challengers can be applied to other optimization techniques like: Ant Colony Optimization, Artificial Bee Colony Optimization etc. to improve their performances. ALC-PSO can be applied efficiently to various applications including Image Processing, Routing in Wireless Sensor Networks, Optical Power Flow etc.

References

[1] Q. Bai, “Analysis of Particle Swarm Optimization Algorithm”, CCSE, no.1, **(2010)** February.

[2] S. S. Aote, “A Brief Review on Particle Swarm Optimization: Limitations & Future Directions, International Journal of Computer Science Engineering (IJCSE).

[3] S. Vijayalakshmi, D. Sudha, S. MercySigamani and K. Kalpana Devi, “Particle Swarm Optimization with Aging Leader and Challenges for Multwaswarm Optimization”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 3, **(2014)** March.

[4] D. P. Rini, S. M. Shamsuddin and S. Yuhaniz, “Particle Swarm Optimization: Technique, System and Challenges”, International Journal of Computer Applications (0975 – 8887), vol. 14, no.1, **(2011)** January.

[5] J. Vina, C. Borra´s and J. Miquel, “Theories of Ageing”.

[6] B. Wang and Z. S. Wu, Z. W. Zhao and H. G. Wang, “Retrieving evaporation duct heights from radar sea clutter using particle swarm optimization algorithm”.

[7] W. N. Chen, J. Zhang, N. Chen, Z. Zhan, H. S. Chung, Y. Li and Y. Shi, “Particle Swarm Optimization with an Aging Leader and Challengers”, ISSN: 949-778X.

Authors

Er. Avneet Kaur, Persuing M. Tech, Computer Science and Engineering from Guru Nanak Dev University, Regional Campus, Jalandhar (India). Has published five papers in journals and conferences.

Er. MandeepKaur, 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 12 years and published seven papers in conferences.