Analytical Experimentation for Constraint Handling on Soft Optimization Technique

Efforts by: Avneet Kaur

Abstract

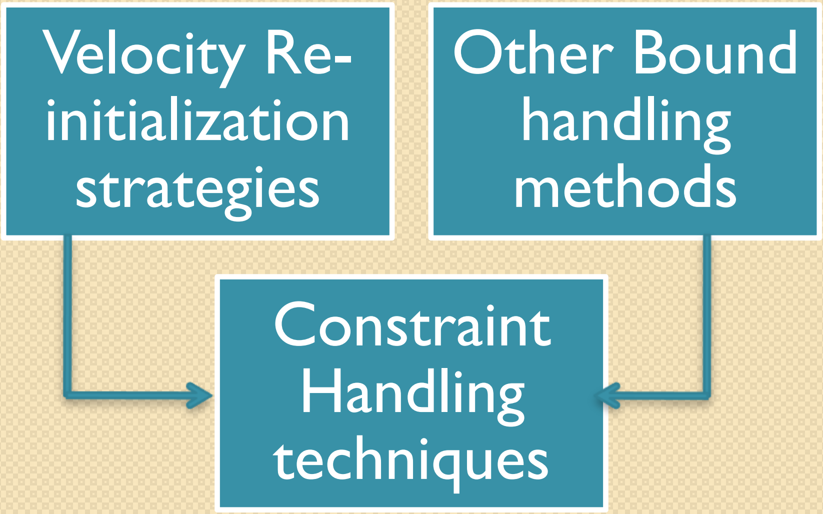
The current work presents the analysis of constraint handling methods on soft optimization techniques including Particle Swarm Optimization and few of its variants by experimenting with different techniques that can be employed for dealing with constraint violations to compare the results given by them.

Constraint Handling

In PSO, constraint handling means applying some boundary restrictions on the particles of the swarm so as to keep the particles within the search bounds for finding the optimal solution to an optimization problem under consideration.

INTRODUCTION

Three standard velocity initialization strategies are used to re-initialize the particles back into search space and other three bound handling methods for dealing with the boundary constraint violations are combined for the experimental analysis of these six strategies for constraint handling on soft computing technique: PSO and ALC-PSO.



Motivation

- Optimization algorithms are widely used in the field of engineering as well as science and technology.

- PSO is a simple and easy optimization technique.

- Many of the variants of PSO have been proposed to improve its performance in one way or other.

- Researching on different variants of PSO is quite interesting.

- Particle Swarm Optimization with Aging Leader and Challengers(ALC-PSO) is a variant of PSO that overcomes stagnation in PSO.

- To further improve the performance of PSO and ALC-PSO, constraint handling methods are applied to them.

Problem Formulation

- The particles tend to leave the boundaries of search space during the iterative process of PSO algorithm.

- Constraint handling is important to enhance the performance and save the time of computation of solution.

- Constraint handling approaches have been applied to the soft optimization technique: PSO and its aging based variant- ALC-PSO to propose more robust algorithms.

Objectives

- To enhance the performance of PSO by implementing its velocity clamped variant.

- The comprehensive comparison of basic PSO and its velocity clamped variant using standard benchmark functions.

- To enhance the performance of ALC-PSO by implementing its velocity clamped variant.

- The comprehensive comparison of standard ALC-PSO with its modified velocity clamped variant.

- Dealing with boundary constraint violations in PSO and ALC-PSO using constraint handling mechanisms.

- To provide comprehensive comparison of PSO and ALC-PSO with different approaches of constraint handling.

LITERATURE SURVEY

For carrying any new research work, it is essential to do a thorough survey of work done in the field. The work done by various authors in the field of PSO, its variants and constraint handling for enhancing its performance has been studied.

Particle Swarm Optimization

Year	Authors	Paper	Main Idea
1995	J. Kennedy and R. C. Eberhart	Particle swarm optimization	1 Optimize non-linear functions 2 Study the relationships of PSO with other optimization algorithms.

Velocity Clamping

Year	Authors	Paper Title	Main Idea
2009	Farrukh Shahzad, A. Rauf Baig, Sohail Masood, Muhammad Kamran	Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO)	Using velocity Clamping to- 1. accelerate the convergence speed of PSO algorithm. 2. avoid premature convergence problem of the algorithm.

Velocity Initialization

Year	Authors	Paper	Main Idea
2012	Andries Engelbrecht	Particle Swarm Optimization: Velocity Initialization	1. Better velocity initialization strategy can enhance the performance of PSO algorithm. 2. Three strategies- initialize to zero, initialize to small values near zero and initialize to random value within domain.

Constraint Handling

Year	Authors	Paper	Main Idea
2012	Nikhil Padhye, Kalyanmoy Deb and Pulkit Mittal	Boundary Handling Approaches in Particle Swarm Optimization	1. Review existing bound handling methods and propose new methods. 2. Testing its performance for different simulation scenarios

PSO with Aging Leader and Challengers

Year	Authors	Paper	Main Idea
2013	Wei-Neng Chen, Jun Zhang, Ni Chen, Zhi-Hui Zhan	Particle Swarm Optimization with an Aging Leader and Challengers	1. Transplanting the aging mechanism to PSO 2. Proposing the PSO with an aging leader and challengers, including leader and challenger particles.

PARTICLE SWARM OPTIMIZATION

PSO simulates the behavior of birds that scatter in search of food. PSO algorithm finds the best solution to a problem from among many possible solutions.

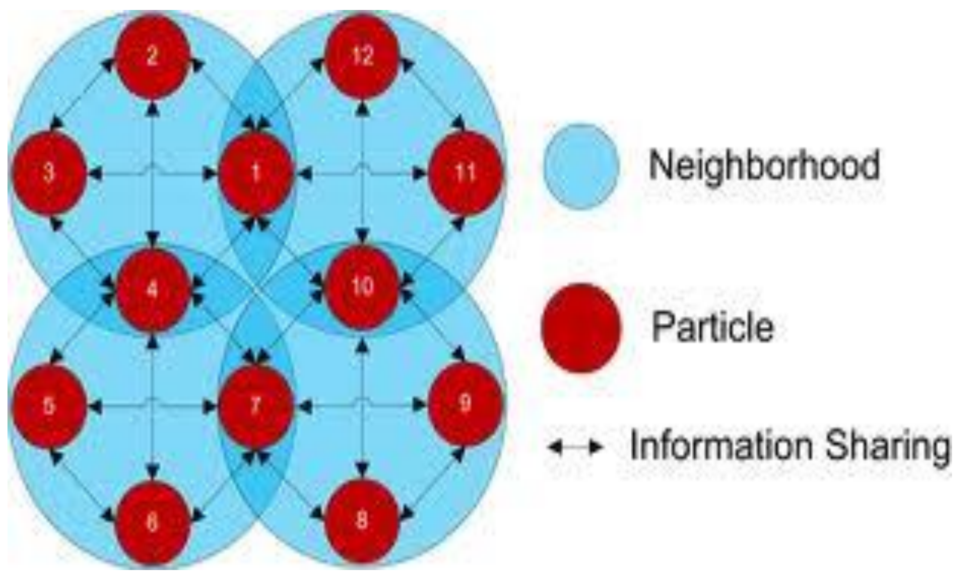
The Term-- Particle Swarm Optimization

- Particle- Every member of the swarm is a particle, which represents a feasible solution to some problem.

- Swarm- Group of particles forms the swarm. All the particles are collectively, called ‘Swarm’.

- Optimization- The process of finding either minimum or maximum value for a problem.

Every particle in PSO, share information with its neighboring particle so as to reach the optimal solution.



Basic PSO Algorithm

PSO is an iterative process. On each iteration in PSO, current velocity of each particle is updated based on three parameters:

i. the particle's current velocity
ii. the particle's local information
iii. global swarm information.

The following velocity update rule is followed:

$$v(t+1) = w * v(t) + c1 * r1 * (p(t) - x(t) + c2 * r2 * (g(t) - x(t))$$

After new velocity is found, new position of particle is found, using equation:

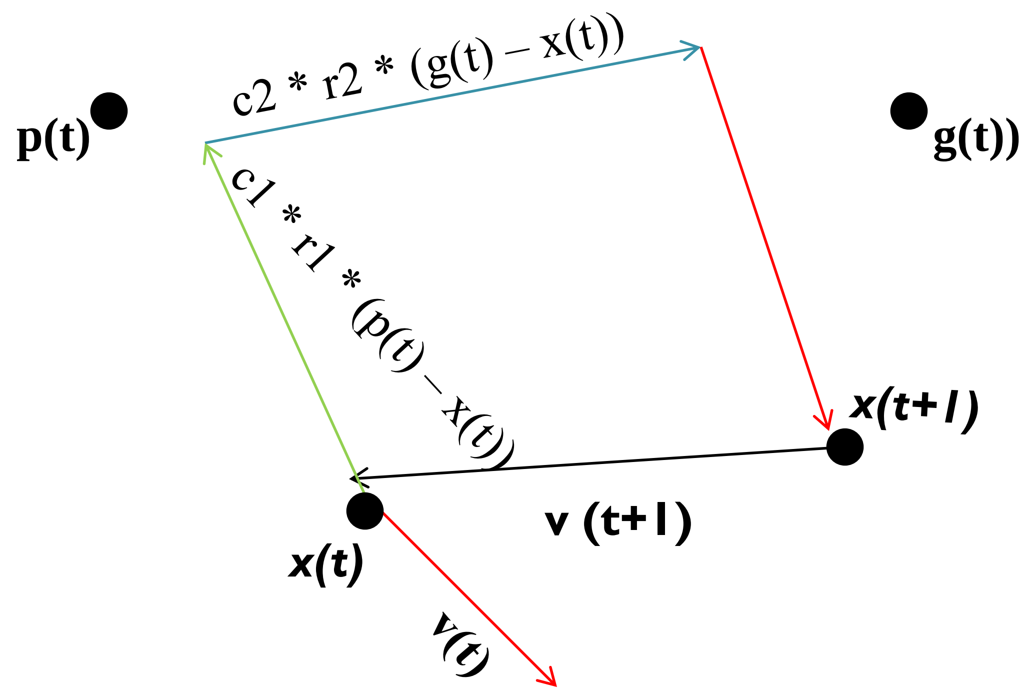
$$x(t+1) = x(t) + v(t+1)$$

$v(t+1) = v(t) + (c1 * r1 * (p(t) – x(t)) + (c2 * r2 * (g(t) – x(t))$

$v(t)$ is the momentum term - The force pulling the particle to continue its current direction

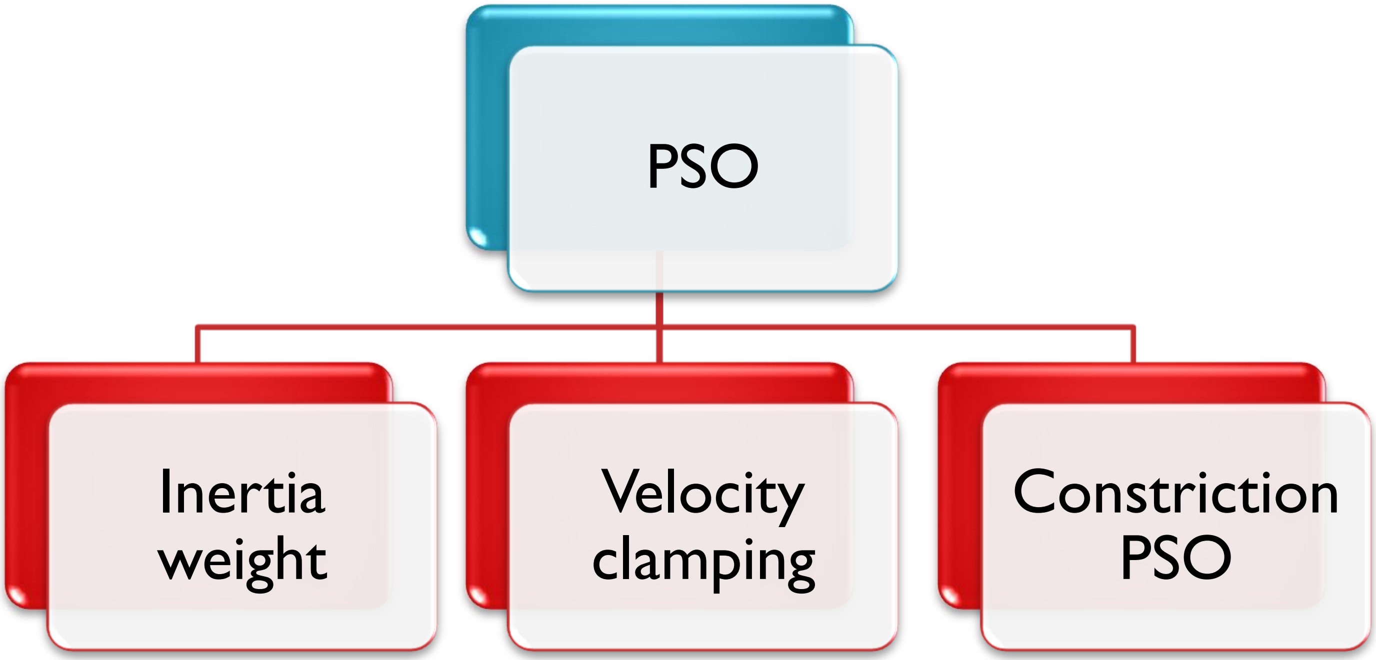
$c1 * r1 * (p(t) – x(t))$ is the cognitive component –It makes the particle to try to return to its personal best position p(t).

$c2 * r2 * (g(t) – x(t)$ is the Social component – It attracts the particle towards the best solution found so far i.e. g(t).



Basic Variants of PSO

Many variants of PSO have been proposed to improve its performance. The basic variants include:



Velocity clamping- Velocity clamping controls the global exploration of particle. If the velocity of particle exceeds the maximum limit then it is set back to the maximum limit allowed.

Inertia Weight- Inertia weight controls the exploration-exploitation ratio of swarm. Inertia weight weighs the contribution of previous velocity and thus, controls the momentum of particle.

Constriction PSO- The Constriction variant balances the exploration and exploitation, and improves the solution found by PSO. It is helpful in achieving convergence.

Stagnation in PSO

The PSO algorithm suffers from stagnation problem i.e. it gets stuck in the local optima and gets converged prematurely.

• If parameters (w, c_1, c_2) are not set correctly, the particles may converge to a point and premature convergence occurs.

• A particle's pbest, gbest positions become same, the particle does not move. All particles tend to accumulate at that point.

• Problem of multiple local optima gives rise to stagnation.

PSO WITH AGING LEADER AND CHALLENGERS (ALC-PSO)

- **Aging-** The phenomenon of decreasing capability of a particle to find the solution with the passage of time is known as 'Aging'.

- **Leader-** The particle selected to lead the entire swarm for finding the best solution of the problem is known as Leader.

- **Challenger-** Another particle of the swarm that challenges the position of leader to itself become the new leader.

- **Lifespan-** The total time period for which the leader will lead the swarm, is known as its 'lifespan'.

Introduction to ALC-PSO

- Several different variants of PSO has been classified, but there is a unique variant of PSO called ALC-PSO.

- It derives the attention because of its unique feature of considering leader for the population to reach to an optimal solution.

- ALC-PSO gives a better solution compared to the PSO.

- For selecting a leader, its lifespan is checked.

In ALC-PSO, each particle's velocity and positions is found using equations:

$$v(t+1) = w*v(t) + c_1*r_1*(p(t) - x(t)) + (c_2*r_2*(Challenger(t)-x(t))$$

$$x(t+1) = x(t) + v(t+1)$$

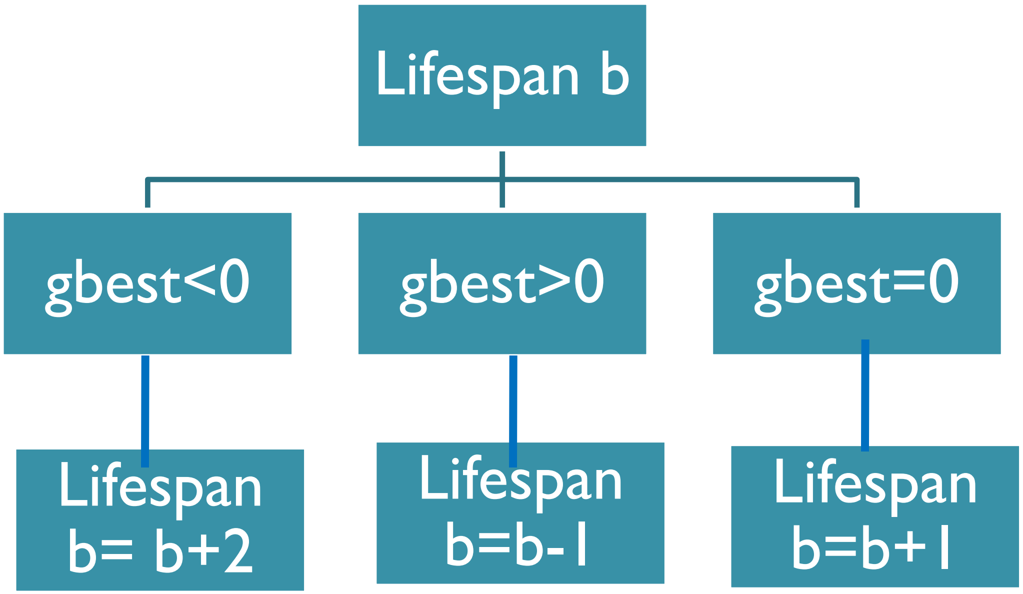
When the lifespan of leader gets exhausted, new leader is selected. The position update rule remains same but the velocity update rule is changed to:

$$v(t+1) = w*v(t) + c_1*r_1*(p(t)- x(t)) + c_2*r_2*(Leader- x(t))$$

Designing and Working Principle of ALC-PSO

The performance of new challenger particle θ is compared with previous leader particle p to decide the new leader of the swarm. The designing of ALC-PSO can be done in three steps:

1. Lifespan Controller - Lifespan b of a particle is adjusted by the lifespan controller. The generated leader checks the gbest and changes the lifespan accordingly:



2. Generation of challenger

- New challenger is generated when the lifespan of the old leader gets exhausted.
- If the performance of particle is greater than the previous leader, the leader is updated.
- When the best solution of the population is found, it is reported.

3. Accepting 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 else the previous leader continues to lead the swarm.

VELOCITY CLAMPING– THE STANDARD BOUNDARY CONSTRAINT

Velocity clamping is the standard boundary constraint. Whenever a particle leaves the boundaries of search space, it is required to be brought back into search bounds.

If velocity>vmax
then velocity = vmax

If velocity < vmin
then velocity = vmin

Reasons for particles leaving search space

•High velocities- Higher velocity values cause the larger values of position of particles, making particles leave the boundaries.

•If the value of inertia weight(w), c_1, c_2 are not correctly set, the velocity may reach infinite value, particles may leave boundaries.

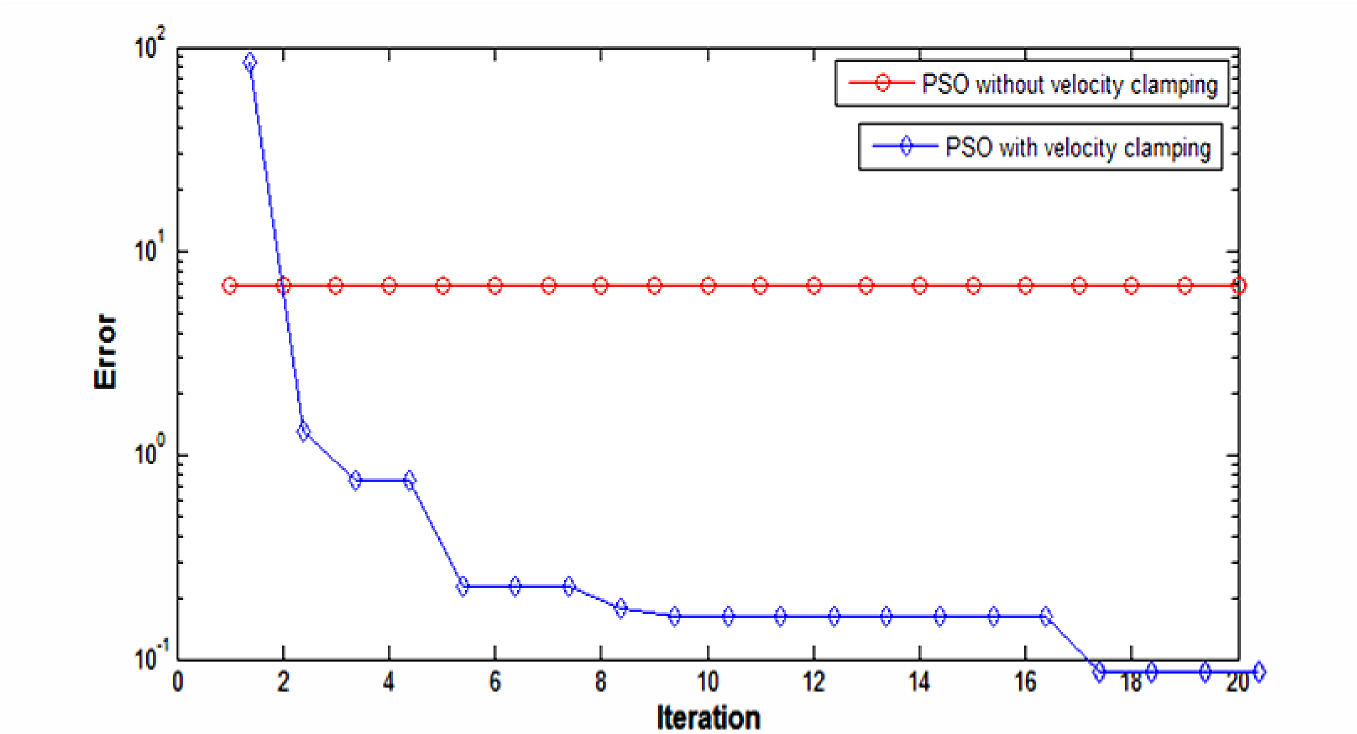
•Large step sizes account for this problem.

Velocity Clamping

- A comparison has been made between ‘with and without velocity clamping’ in PSO and ALC-PSO to show its importance and an improvement in performance on applying this boundary constraint.

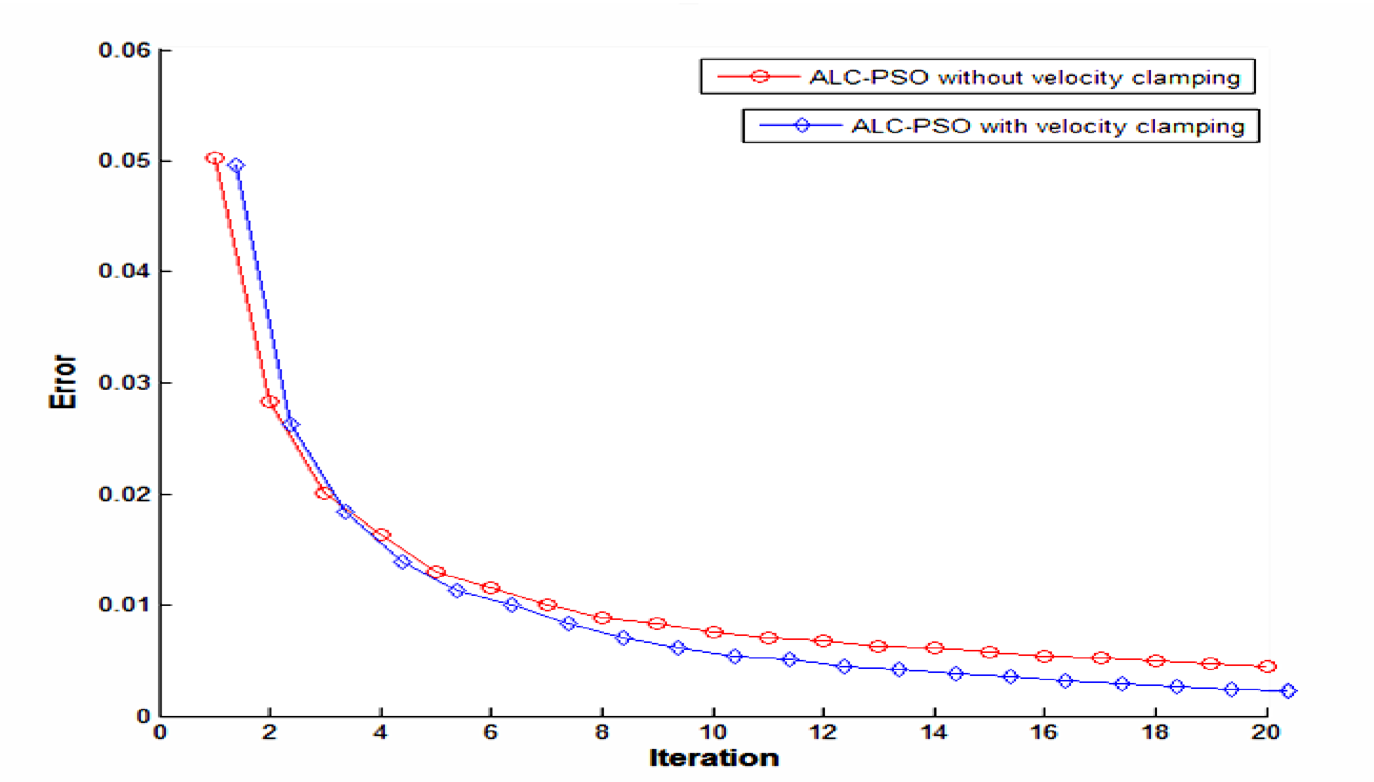
- To see better variation of results, two curves showing the error plots by using velocity clamping and without velocity clamping are presented.

Comparison of Basic PSO and Velocity clamped PSO



gbest of Basic PSO	gbest of Velocity Clamped PSO
0.9273	0.0550

Comparison of Basic ALC-PSO and Velocity clamped ALC- PSO



gbest of Basic ALC-PSO	Gbest of Velocity clamped ALC-PSO
0.0894	0.0850

Effects of applying Velocity clamping

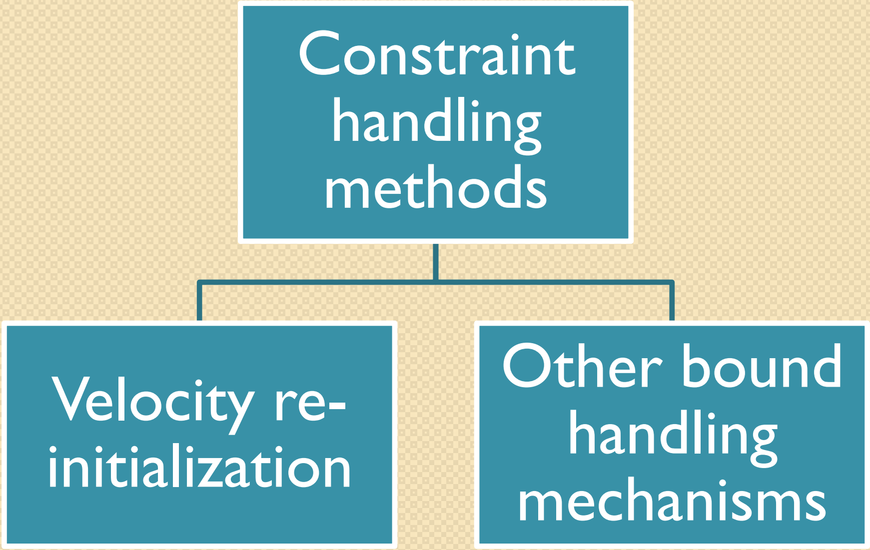
- Decreases the step size for efficient performance.

- No negative effect on the position of particles.

- Changes particle’s search direction to have better exploration.

CONSTRAINT HANDLING

It is essential to implement some constraint handling mechanisms to bring the particles back into search bounds, for the better performance of PSO algorithm and its variants.



Introduction

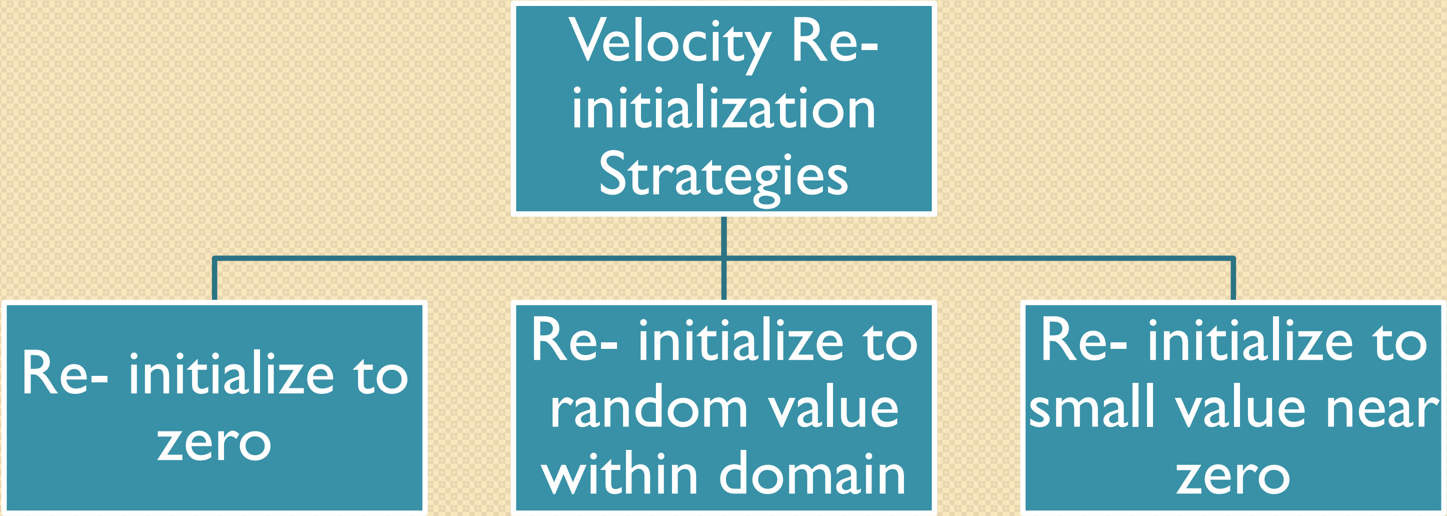
- **Boundary constraints-** The search space is restricted with upper and lower limits within which the solution is found. These are known as boundary constraints.

- **Boundary constraint violation-** If a particle leaves the search space, it violates the boundary constraints. These particles are required to be brought back into the search space.

- **Constraint Handling-** The methods employed for dealing with the boundary constraint violations.

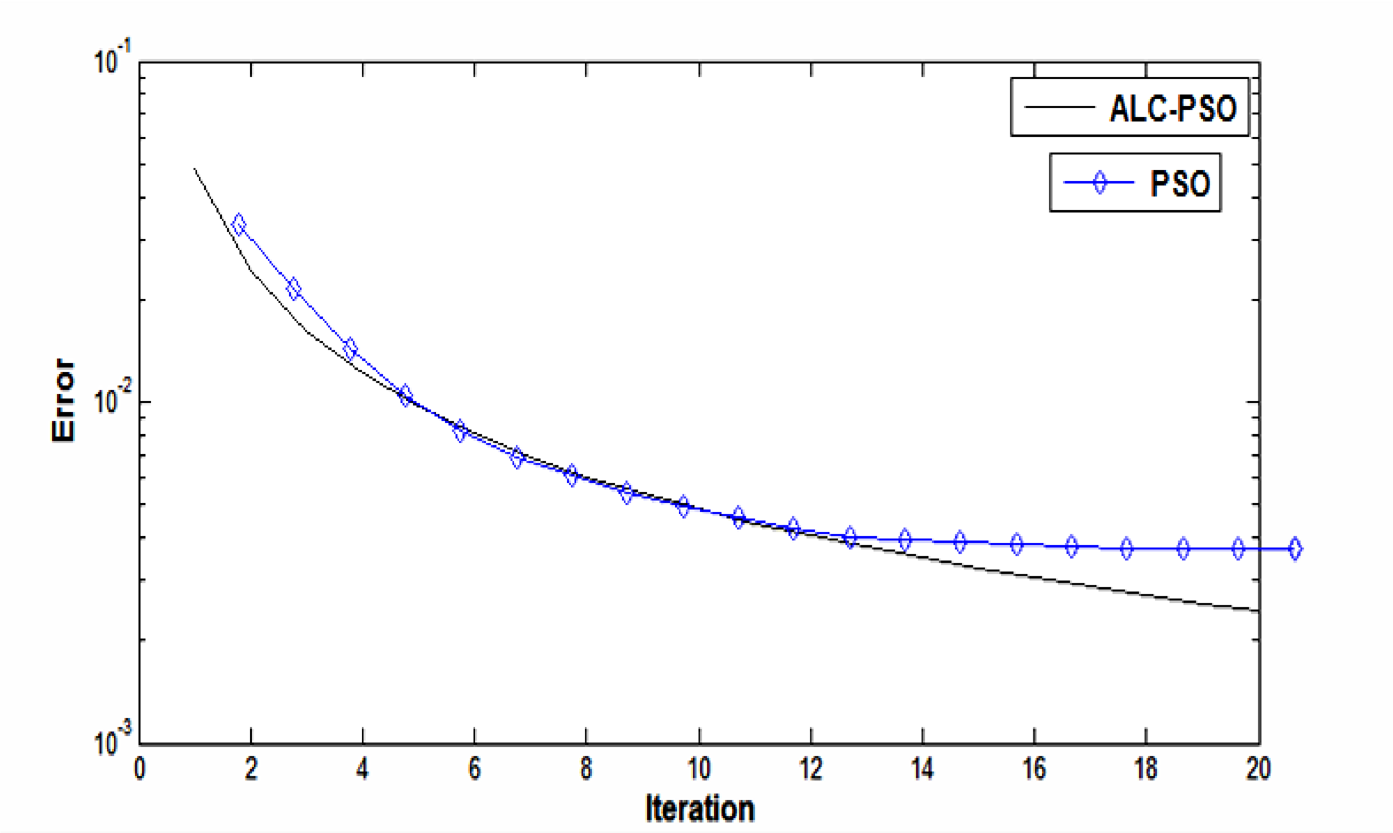
IMPLEMENTING VELOCITY RE-INITIALIZATION STRATEGIES

If any particle leaves the boundaries of the search space or its velocity increases its pre-defined value, it needs to be re-initialized so that it is brought back into the search space.



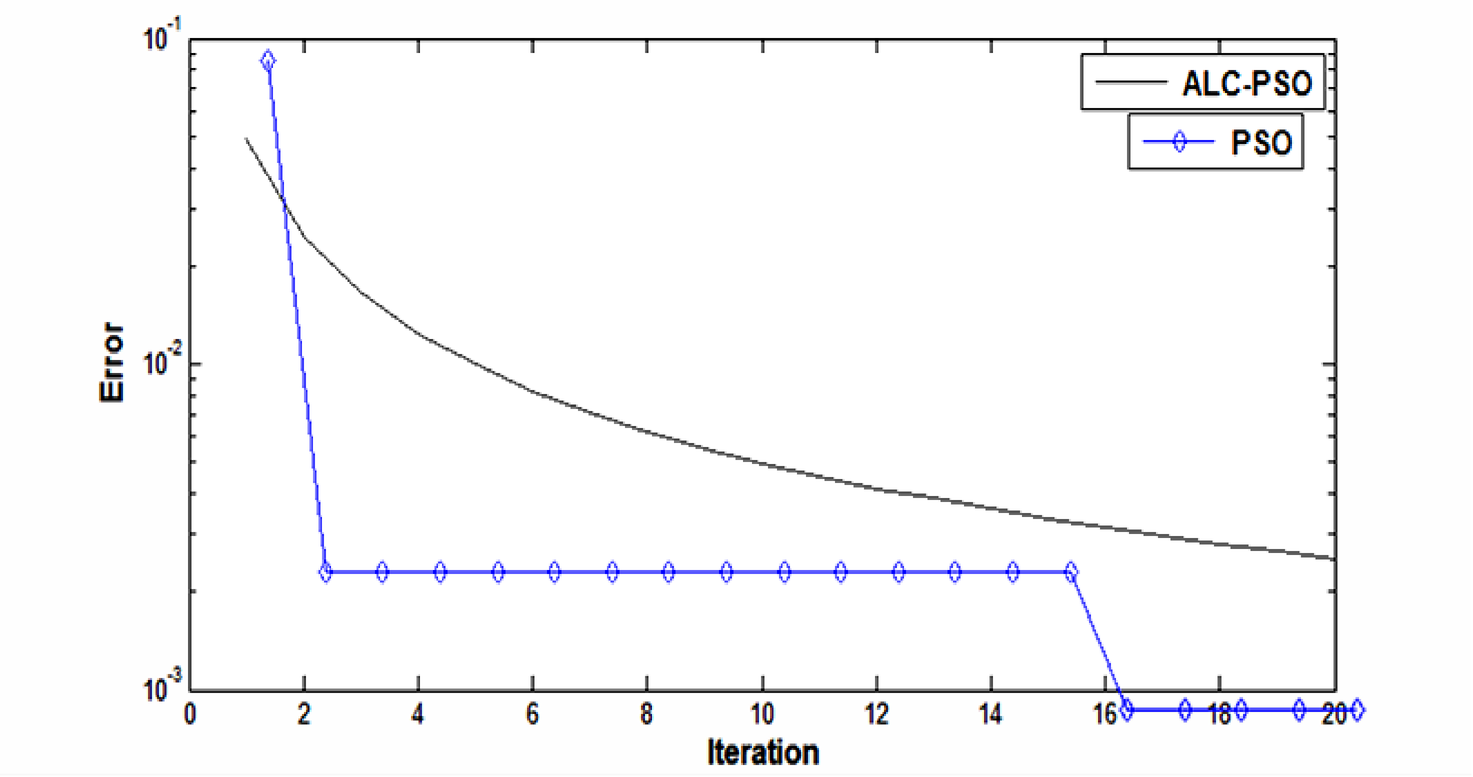
Velocity re-initialization to 0

Bringing the particle back to initial momentum, by re-initializing to zero velocity.



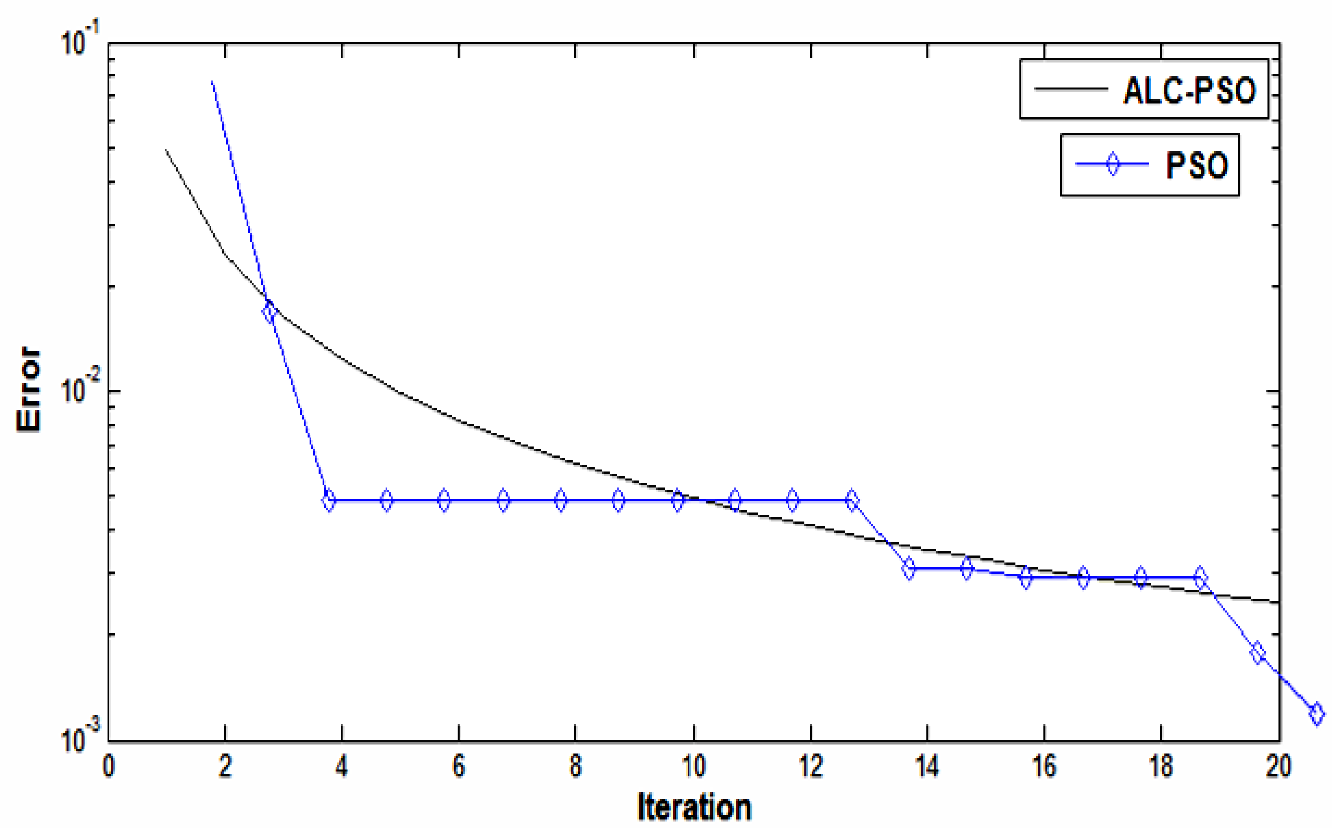
Velocity re-initialization to value within domain

The velocity of the particle is set to some random value within a pre-defined range. The range taken here is [-vmax,vmax]



Velocity re-initialization to small random value near zero

The velocity of the particle is re-initialized to some small random value near to 0. Values taken here are 0.004, 0.002.

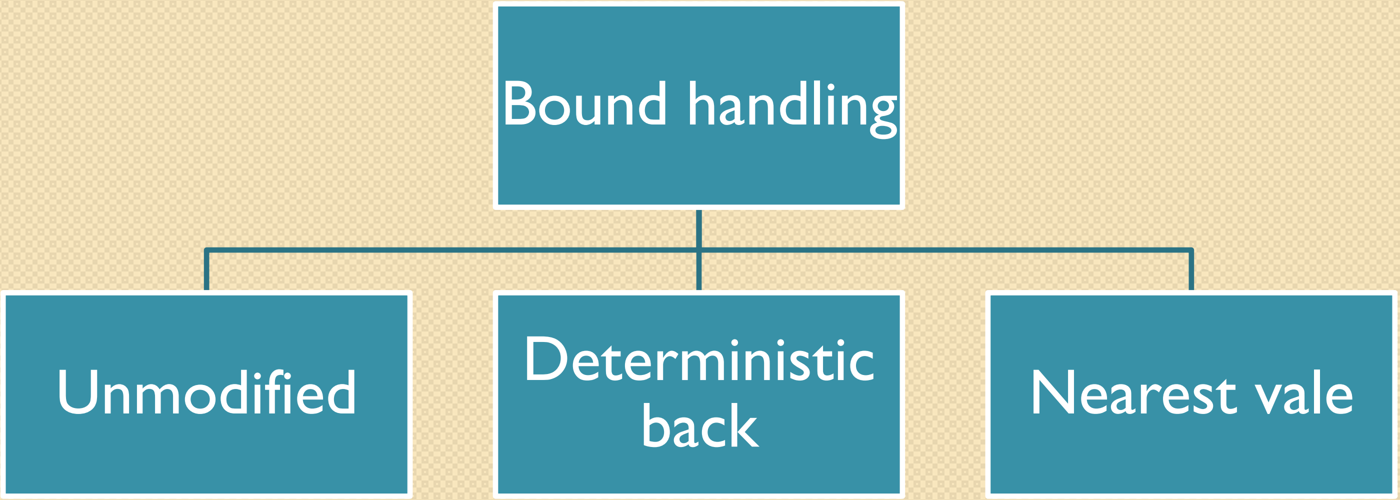


Comparing performance of PSO and ALC-PSO with different velocity re-initialization strategies

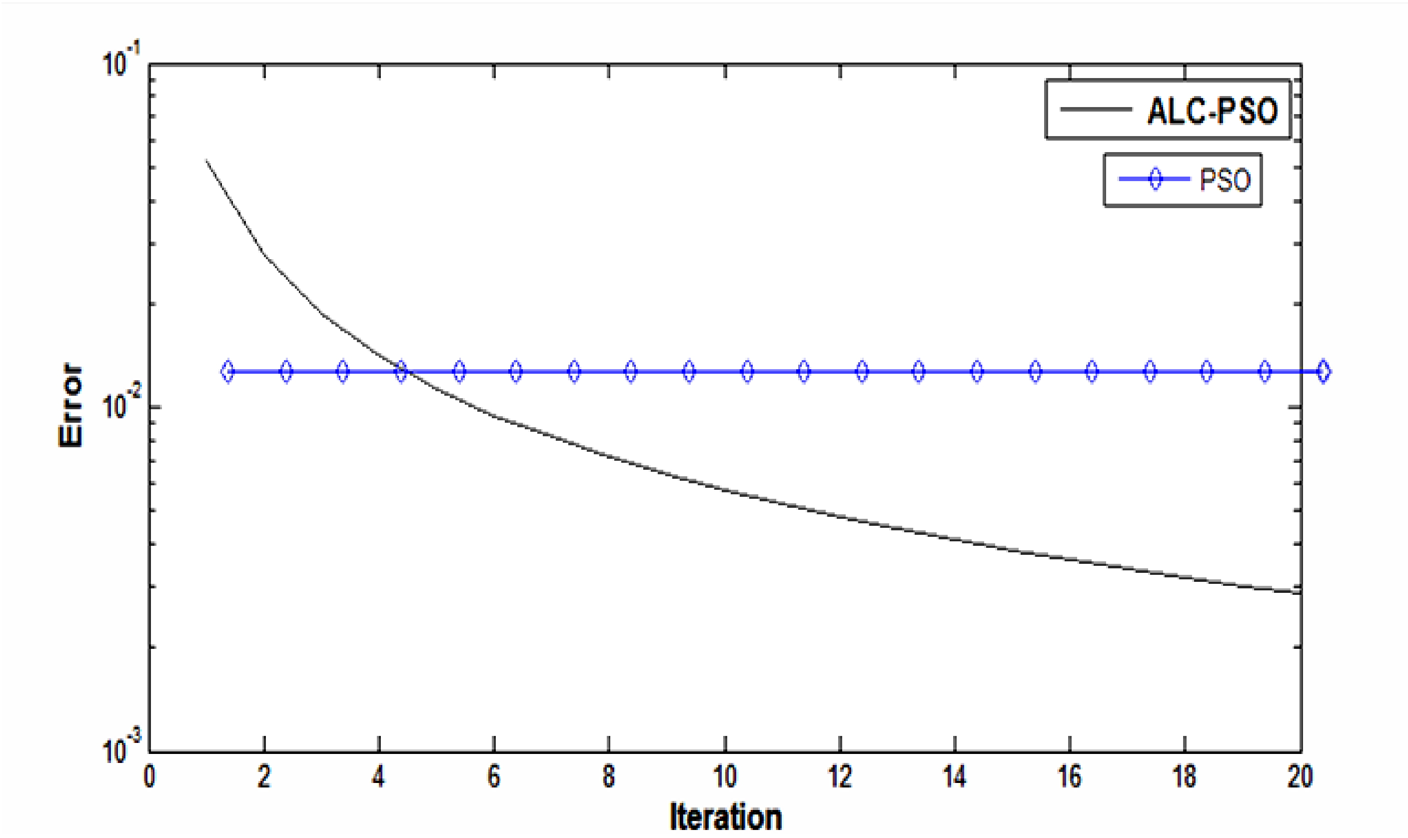
Velocity Re-Initialization	gbest value for PSO	gbest value for ALC-PSO
Zero re- initialization	0.0880	0.0488
Random value re-initialization	0.0196	0.0505
Re-initialization to small value near zero	0.6420	0.0493

VELOCITY BOUND HANDLING

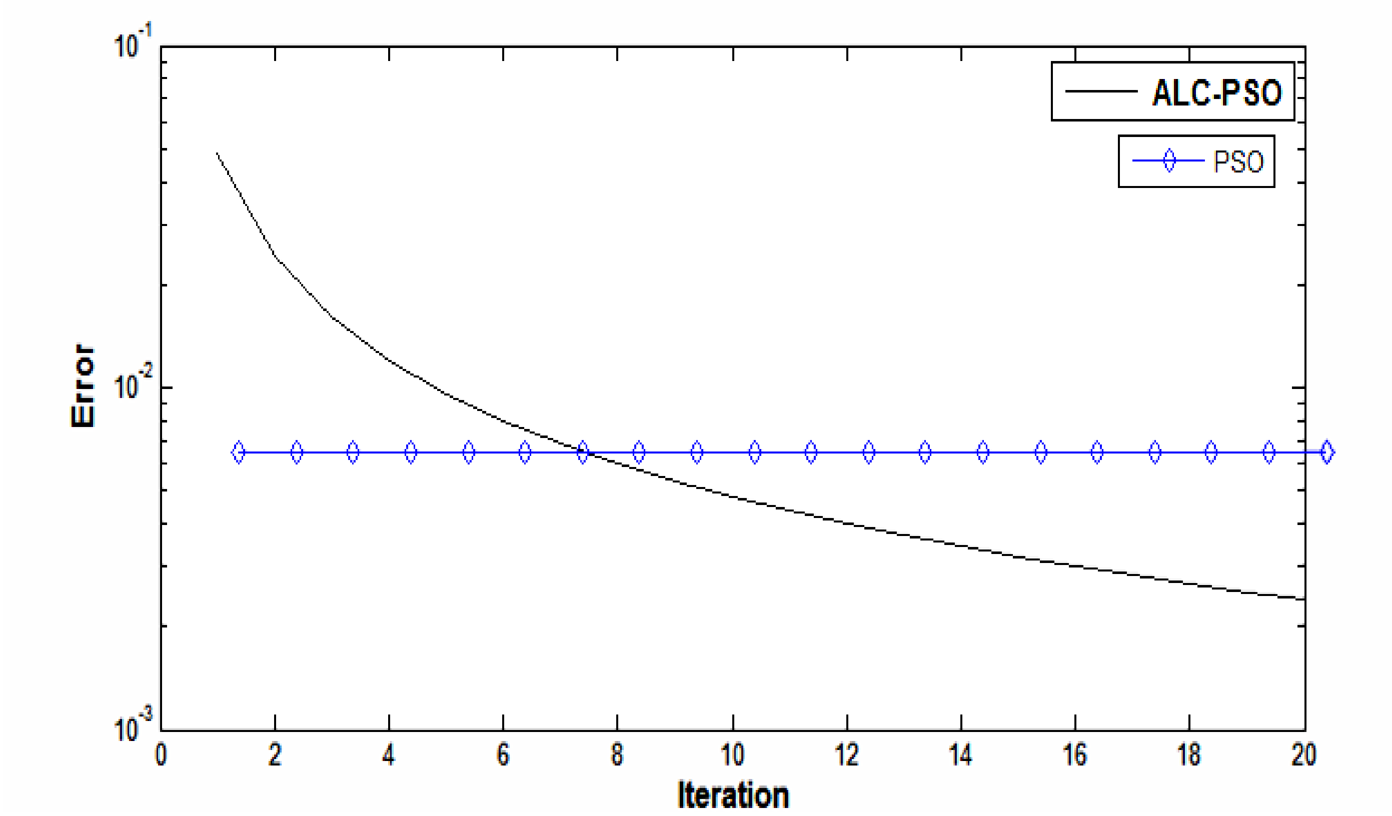
- Creation of feasible-only solutions during the evolutionary search.
- Explicit mechanism to repair an infeasible solution i.e. bringing the infeasible solution back into the feasible search space.



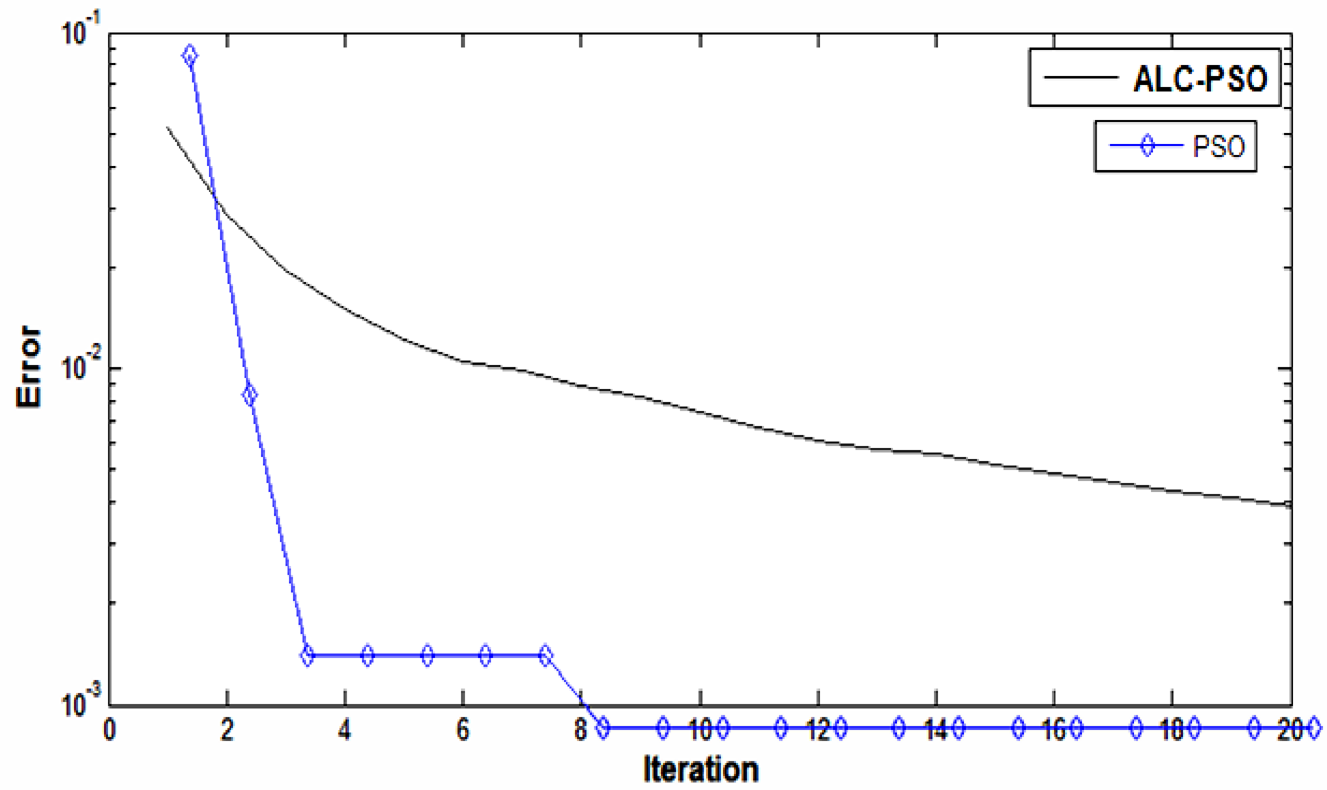
Unmodified - The velocity of the particle leaving boundaries of search space is unaltered



Deterministic back- A constant negative term k is multiplied with the velocity of the particle. The constant is taken to be 0.5 here.



Nearest value- The velocity of the particle is set to the peak value assigned for the process.



Comparing performance of PSO and ALC-PSO with different bound handling strategies

Bound Handling	gbest for PSO	gbest for ALC-PSO
Unmodified	0.4143	0.0574
Deterministic back	0.1601	0.0481
Nearest Value	0.0263	0.0779

Comparing the results of these six strategies for Ackley function

Constraint Handling Technique	gbest for PSO	gbest for ALC-PSO
Zero re-initialization	0.0880	0.0488
Random re-initialization	0.0196	0.0505
Re-initialization to small value near zero	0.6420	0.0493
Unmodified method	0.4143	0.0574
Deterministic method	0.1601	0.0481
Nearest method	0.0263	0.0779

Results of applying Constraint handling techniques

After comparing the results for five benchmark functions, following conclusions can be made-

- Unmodified technique gives worst results.
- Zero re-initialization and small value near zero initialization gives similar results.
- Random re-initialization within domain is not a good technique.

Conclusion

- Velocity clamping is essential to restrict the particles from leaving the boundaries of search space.

- Improvement in performance can be seen when bound handling approaches are applied to the algorithms.

- The velocity re-initialization to zero strategy works best among different strategies for both algorithms.

- The best approach for velocity bound handling is found to be deterministic back for ALC-PSO while for PSO, nearest method works well.

Future Scope

- Initializing velocities of particles in ALC-PSO using a uniform distribution technique within search space, such that every particle in the swarm gets at least one neighbor particle, which is having better fitness value compared to its own to improve its performance.

- Introducing an adaptive velocity mechanism, which increases the velocity of a particle if some progress is found in recent iterations and decrease velocity if no progress is observed during recent iterations.

Thankyou