Assignment No: 06

Aim: Implement Particle swarm optimization for benchmark function (eg. Square, Rosenbrock function). Initialize the population from the Standard Normal Distribution. Evaluate fitness of all particles.

Use:

c1=c2 = 2

Inertia weight is linearly varied between 0.9 to 0.4.

Global best variation

Objectives:

1. To learn swarm algorithm
2. To learn about optimization algorithm.

Software Requirements:

Ubuntu 18.04

Hardware Requirements:

Pentium IV system with latest configuration

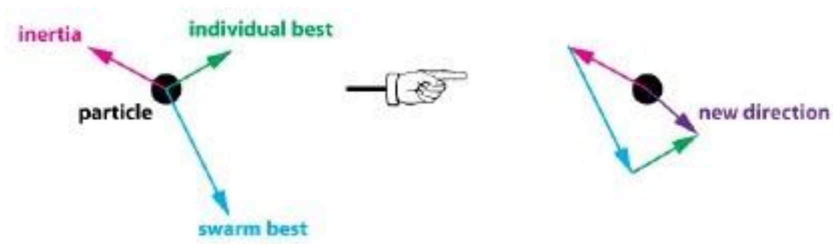
Theory:

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

PSO uses a bunch of particles called *the swarm*. These particles are allowed to move around & explore the search-space.

These particles move in a direction which is guided by —

1. The particle's own previous velocity (*Inertia*)
2. Distance from the individual particles' best known position (*Cognitive Force*)
3. Distance from the swarms best known position (*Social Force*)



Particle movement is overpowered by the swam direction

Essentially the particles collectively communicate with each other to converge faster. The swarm doesn't fully explore the search space but potentially finds a better solution.

Interestingly the overall direction of the swarm movement can be changed at any point of time when a particle's individual best is better than the swarm best. This allows a lot of *disorder* and more chances of getting close to the global minima of the cost function

Algorithm

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) (a)

present[] = persent[] + v[] (b)

v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually

c1 = c2 = 2.

The pseudo code of the procedure is as follows

For each particle

Initialize particle

END

Do

For each particle

Calculate fitness value

If the fitness value is better than the best fitness value (pBest) in history

set current value as the new pBest

End

Choose the particle with the best fitness value of all the particles

as the gBest For each particle

Calculate particle velocity

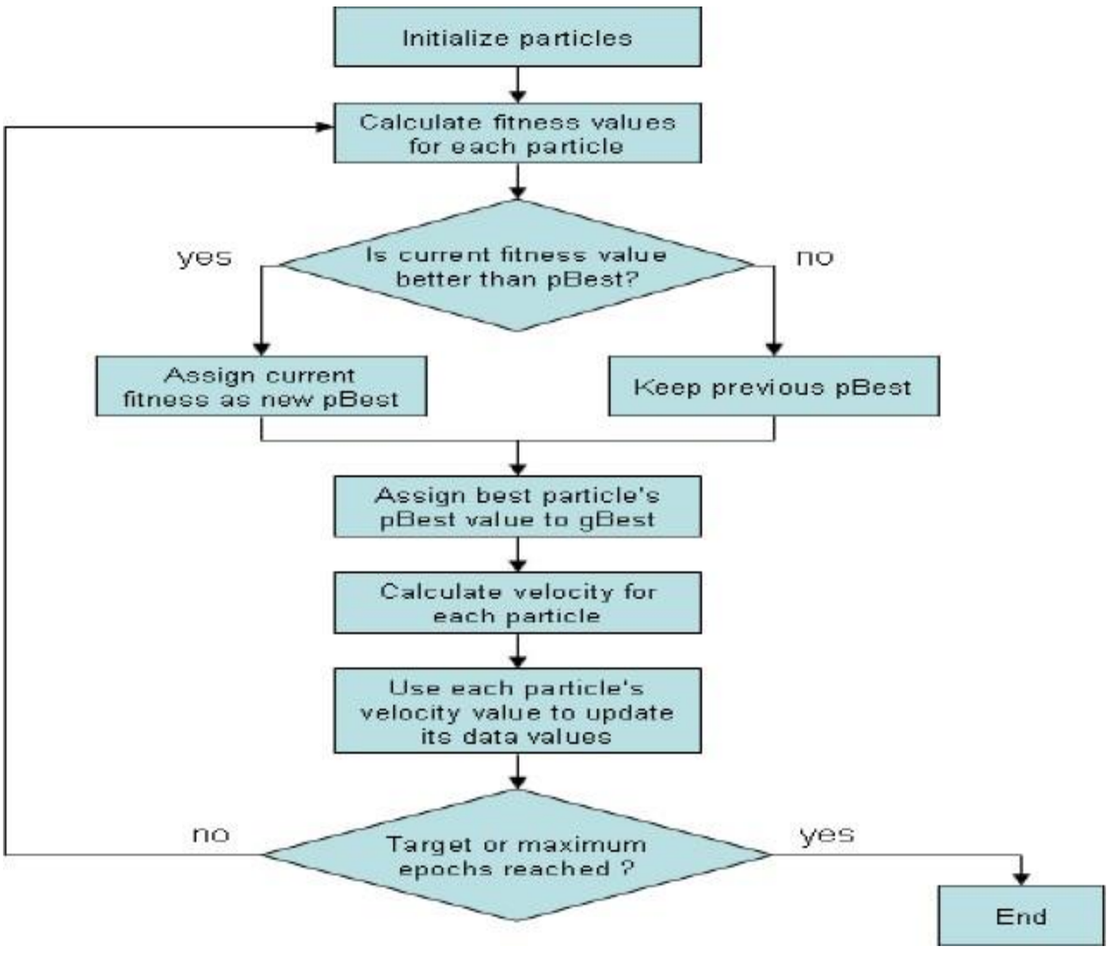
according equation (a) Update

particle position according equation

(b) End

While maximum iterations or minimum error criteria is not attained

Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.



Conclusion: Thus we learnt implementation of particle swarm optimization for benchmark function.