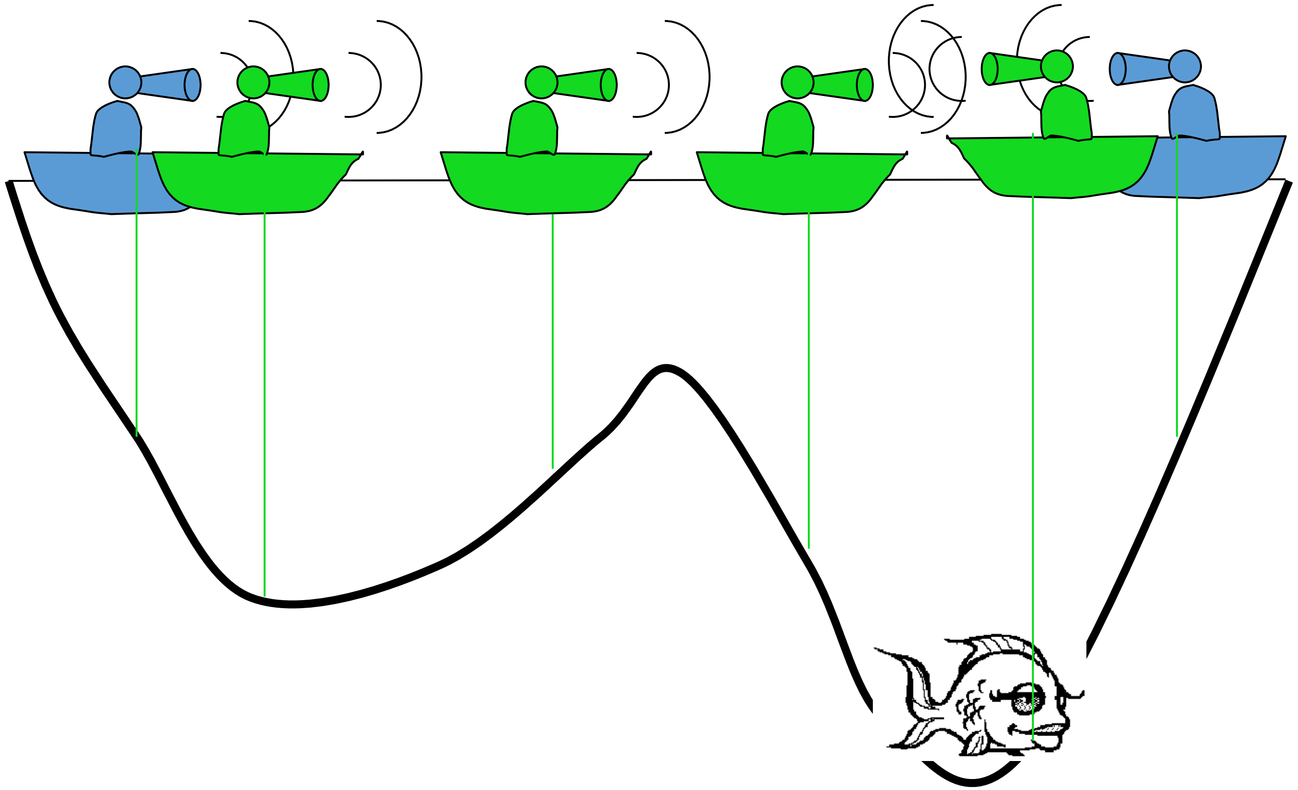
Particle Swarm Optimization

Abhishek Roy

Friday Group Meeting

Date: 05.25.2016

Cooperation example

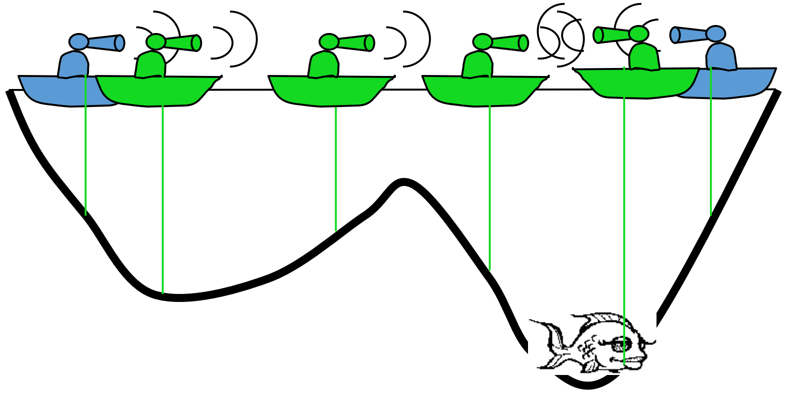


Basic Idea

- ❑ PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms.

- ❑ It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution.

- ❑ Each particle is treated as a point in a N-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles.



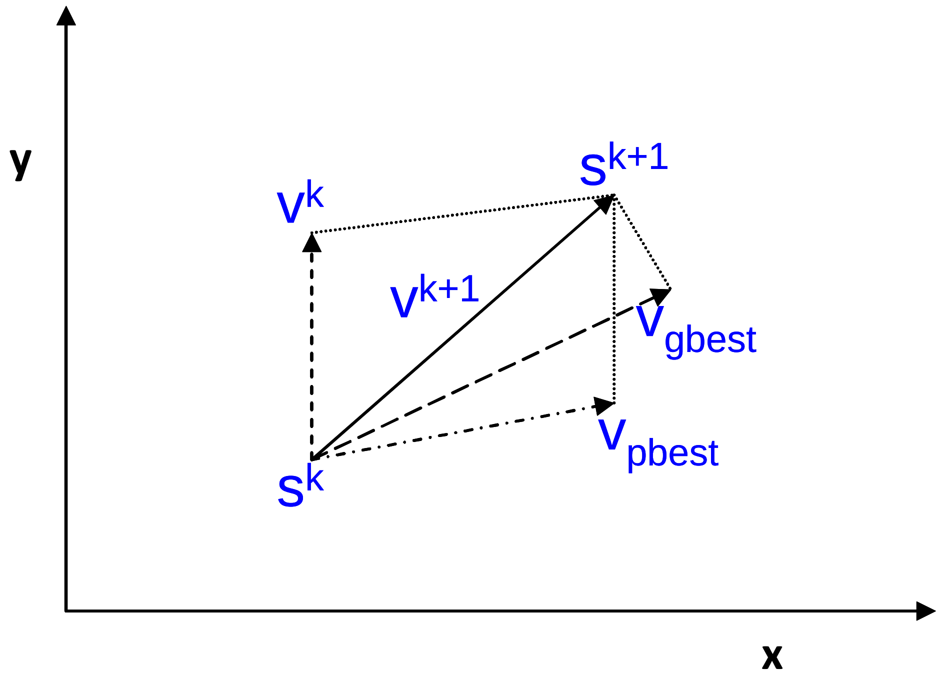
Basic Idea

- Each particle keeps track of its coordinates in the solution space which are associated with the best solution (fitness) that has achieved so far by that particle. This value is called personal best , ***pbest***.

- Another best value that is tracked by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called ***gbest***.

- The basic concept of PSO lies in accelerating each particle toward its *pbest* and the *gbest* locations, with a random weighted acceleration at each time step.

Basic Idea



Concept of modification of a searching point by PSO

s^k: current searching point.
s^{k+1}: modified searching point.
v^k: current velocity.
v^{k+1}: modified velocity.
v_{pbest} : velocity based on pbest.
v_{gbest} : velocity based on gbest

Particle Swarm Optimization (PSO)

- Each particle tries to modify its position using the following information:
  - the current positions,
  - the current velocities,
  - the distance between the current position and pbest,
  - the distance between the current position and the gbest.

$$\mathbf{V_i^{k+1} = \mathit{w}\, V_i^k + c_1\, rand_1(\ldots) \times (\, pbest_i - s_i^k\,) + c_2\, rand_2(\ldots) \times (\, gbest - s_i^k\,)} \qquad \mathbf{(1)}$$

v_i^k : velocity of agent i at iteration k,
w: weighting function,
c_j : weighting factor,
rand : uniformly distributed random number between 0 and 1,
s_i^k : current position of agent i at iteration k,
$pbest_i$: pbest of agent i,
gbest: gbest of the group.

Particle Swarm Optimization (PSO)

The following weighting function is usually utilized in (1)

$$\boldsymbol{w = \text{wMax}-[(\text{wMax}-\text{wMin}) \times \text{iter}]/\text{maxIter}} \qquad \textbf{(2)}$$

where, wMax= initial weight,

wMin = final weight,

maxIter = maximum iteration number,

iter = current iteration number.

$$s_i^{k+1} = s_i^k + V_i^{k+1} \qquad \textbf{(3)}$$

Particle Swarm Optimization (PSO)

Comments on the Inertial weight factor:

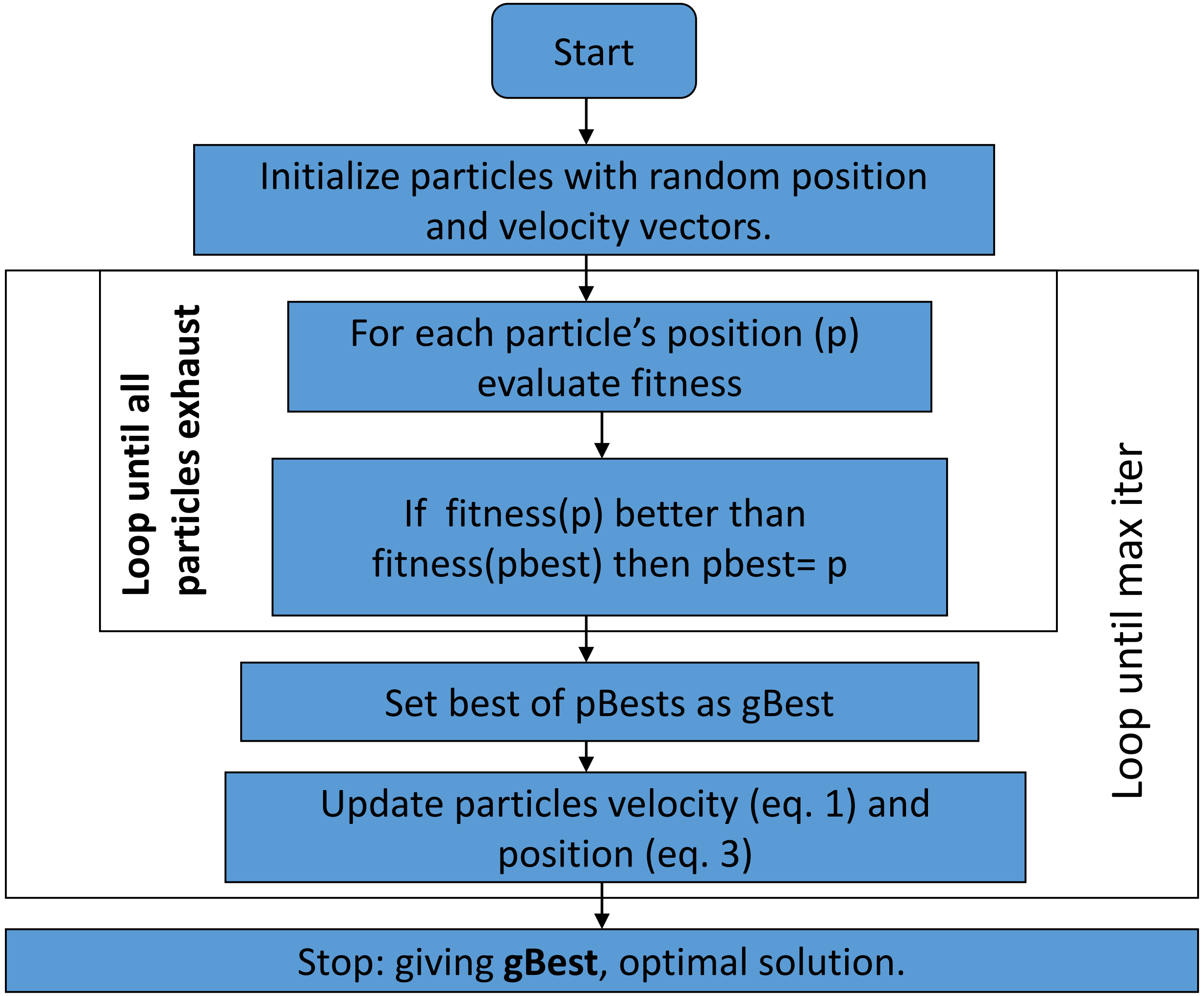
A large inertia weight (*w*) facilitates a global search while a small inertia weight facilitates a local search.

By linearly decreasing the inertia weight from a relatively large value to a small value through the course of the PSO run gives the best PSO performance compared with fixed inertia weight settings.

Larger w ----------- greater global search ability
Smaller w ------------ greater local search ability.

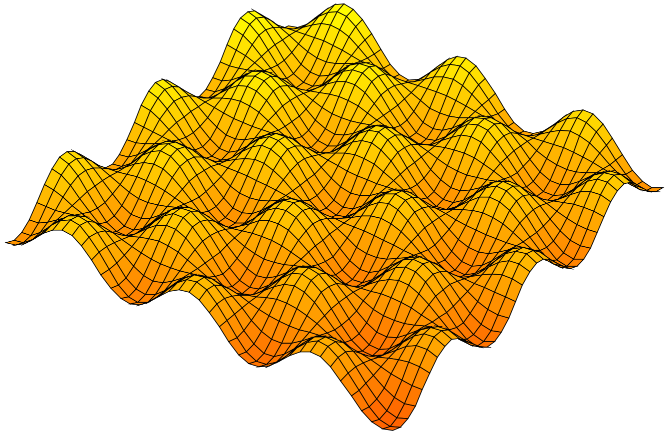
Particle Swarm Optimization (PSO)

Flow chart depicting the General PSO Algorithm:

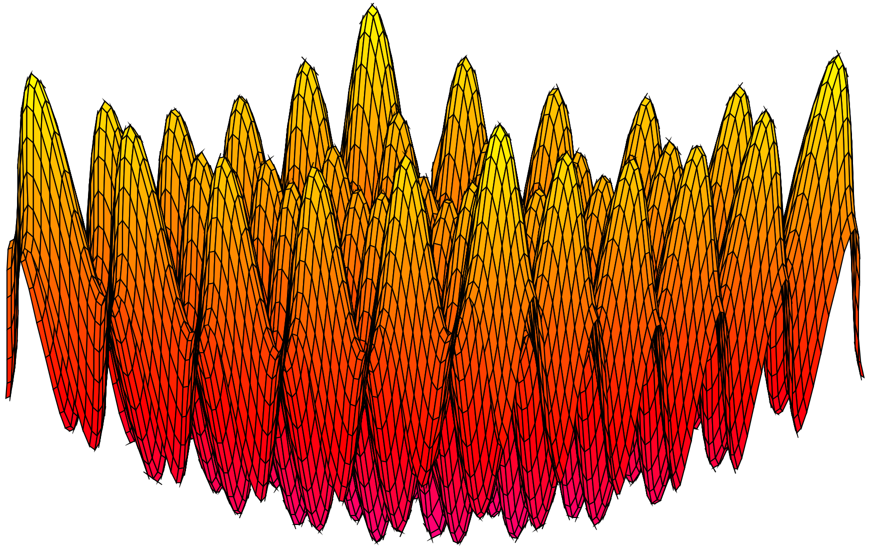


Some functions often used for testing real-valued optimization algorithms

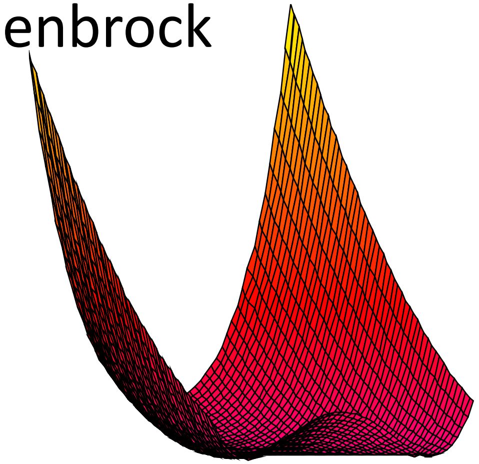
Griewank



Rastrigin



Rosenbrock



… and some typical results

Optimum=0, dimension=30

Best result after 40 000 evaluations

30D function	PSO Type 1"	Evolutionary algo.(Angeline 98)
Griewank [±300]	0.003944	0.4033
Rastrigin [±5]	82.95618	46.4689
Rosenbrock [±10]	50.193877	1610.359

Variants of PSO

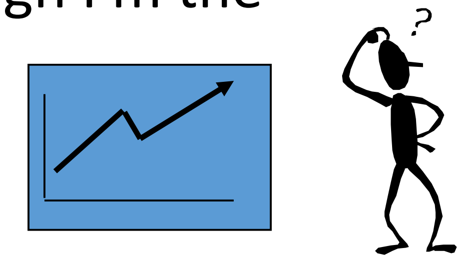
- Discrete PSO ……………… can handle discrete binary variables

- MINLP PSO………… can handle both discrete binary and continuous variables.

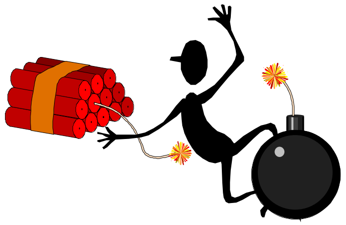
Adaptive swarm size

There has been enough improvement

although I'm the worst

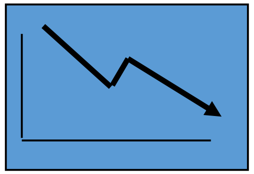


I try to kill myself



I'm the best

but there has been not enough improvement



I try to generate a new particle

Adaptive coefficients

av

$rand(0...b)(p-x)$

The better I am, the more I follow my own way

The better is my best neighbour, the more I tend to go towards him

Constrained PSO

- ❑ Form a Lagrange multiplier or a some other penalty function

- ❑ Apply PSO

Thank You