CBCT: CS 535

Particle Swarm Optimization

Lecture 19

Zubin Bhuyan

Department of CSE,
Tezpur University
http://tezu.ernet.in/~zubin

Swarm Intelligence

- ***Swarm intelligence***
  - collective behavior of simple rule-following agents
  - overall behavior of the entire system appears intelligent
- In Nature such behavior is seen in bird flocks, fish schools, ant colonies and animal herds
- ***Particle Swarm Optimization*** is a class of stochastic, population based optimization techniques

Particle Swarm Optimization

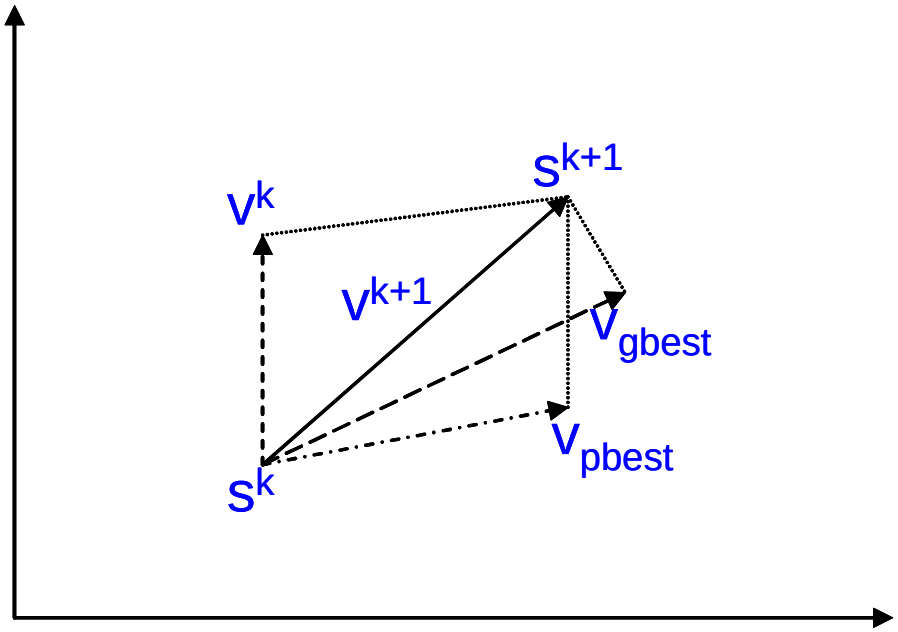
- PSO was developed in 1995 by James Kennedy (*social-psychologist*) and Russell Eberhart (*electrical engineer*). †
- PSO is inspired from the concept of social interaction and is used for problem solving.
- A swarm of *n* agents or *particles flies* around in the search space looking for the best solution
  - *Particles* communicate directly or indirectly with one another to determine its search direction.

† Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942-1948, IEEE Press.

PSO

- *pBest*: Best value obtained so far by an individual particle
  - Each particle has its own *pBest*
- *gBest*: Best of all *pBest*
- The basic concept of PSO is to accelerate each particle toward its *pBest* and the *gBest* locations
  - Usually with a random weighted acceleration at each time step

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

Concept of modification of a searching point by PSO

ŧ Slide taken from *Varadarajan Komanduri, Research Assistant, ECE Dept.,Villanova University*
http://www23.homepage.villanova.edu/varadarajan.komanduri/PSO_meander-line.ppt

PSO Algorithm

1. Initialize a population of particles randomly over a problem space with random velocities.
2. Evaluate fitness of each particle.
3. If current fitness of particle is better than *pbest*, then set *pbest* value equal to current fitness. Set *pbest* location to current location.
4. If current fitness is better than *gbest*, reset *gbest* to current fitness value. Set new *gbest* location to current location.
5. Change velocity according to the equation:
   $v_{vid} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id})$
6. Change the position according to equation:
   $x_{id} = x_{id} + v_{id}$
   Here w is inertia weight, c_1 and c_2 are acceleration constants, and rand() is a random number generator function.
7. Loop back to Step 2 until end criterion is satisfied, or maximum number of iterations is completed.

PSO Algorithm

- $v_{id} = w * v_{id}$

  $+ c_1 * rand() * (p_{id} - x_{id})$

  $+ c_2 * rand() * (p_{gd} - x_{id})$

  w: Inertia weight of current velocity

  c_1 : Acceleration component of *cognitive* part

  c_2 : Acceleration component of *social* part

- $x_{id} = x_{id} + v_{id}$

Thanks!