Lecture Series on

Intelligent Control

Lecture 26
Particle Swarm Optimization

Kwang Y. Lee
Professor of Electrical and Computer Engineering
Baylor University
Waco, TX 76798, USA
Kwang_Y_Lee@baylor.edu

Outline

1. Introduction
2. Basic Particle Swarm Optimization
3. Variations of Particle Swarm Optimization Techniques
4. Parameter Selections and Constriction Factor Approach
5. Research Areas and Applications
6. Conclusions

1. Introduction

From 80's

Swarm behavior research based on Artificial Life research

Craig W. Reynolds developed ***boid**** as a swarm model with simple rules and generated complicated swarm behavior by CG animation. He uses the following ***three vectors*** as simple rules:

(1) to step away from the nearest agent
(2) to go toward the destination of the swarm
(2) to go to the center of the swarm

Complicated swarm behavior can be represented by ***simple rules (combination of vectors)***

*The name "boid" corresponds to a shortened version of "bird-oid object", which refers to a bird-like object.

C. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," 1987 Computer Graphics, vol.21, no. 4, pp. 25-34.

Psychology

Boyd and Richerson examined the decision process of human being and developed the concept of ***individual learning and cultural transmission***. People utilize the following two important kinds of information in decision process:

(1) Their own experience

They have tried the choices and know which state has been better so far, and they know how good it was.

(2) Other people's experiences

They have knowledge of how other agents around them have performed. Namely, they know which choices their neighbors have found are the most positive so far and how positive the best pattern of choices was.

R. Boyd and P. Richerson, *Culture and the Evolutionary Process*, University of Chicago Press, 1985.

Psychology

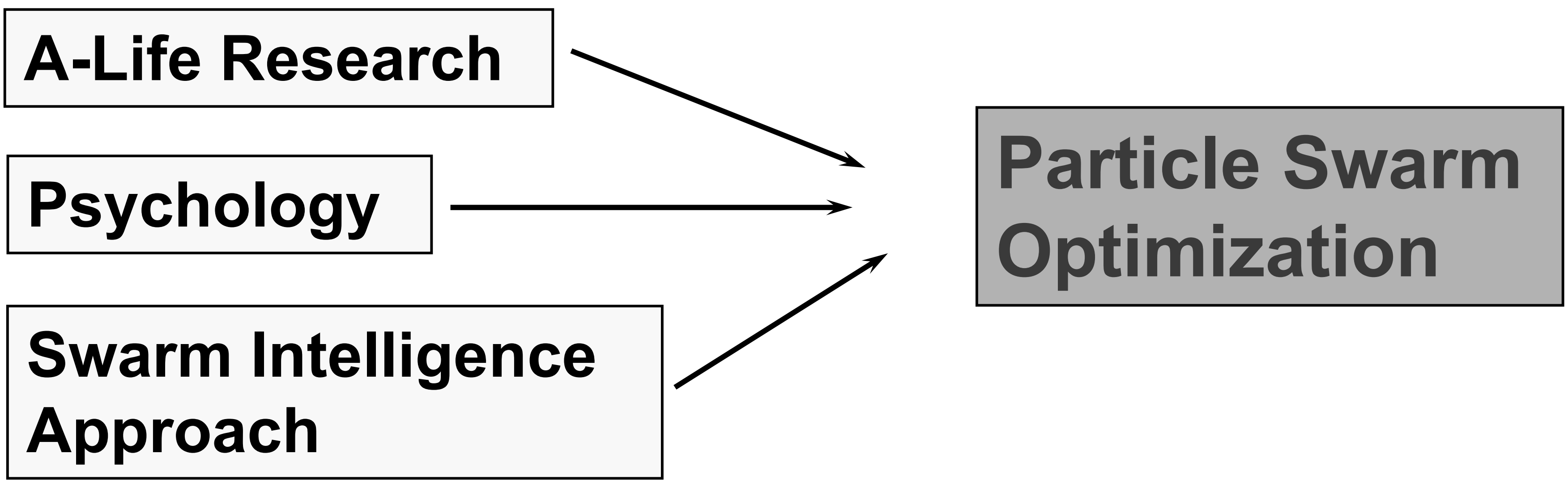
Each agent decides his decision using
his own experience and ***other peoples' experiences***.

From the beginning of 90's

Swarm Intelligence Approach

Colorni, Dorigo and Maniezzo developed ***Ant Colony Optimization (ACO)*** mainly based on the social insect, especially ***Ant***, behavior.

➡ Swarm behavior can be used for optimization



A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies", *Proc. of First European Conference on Artificial Life*, pp.134-142, Cambridge, MA: MIT Press 1991.

2. Basic Particle Swarm Optimization

Gbest model

- Optimization technique with ***continuous*** variables developed through simulation of simplified social models such as swarm of birds.

* Realizes ***Stochastic and multipoint search*** like Genetic Algorithm

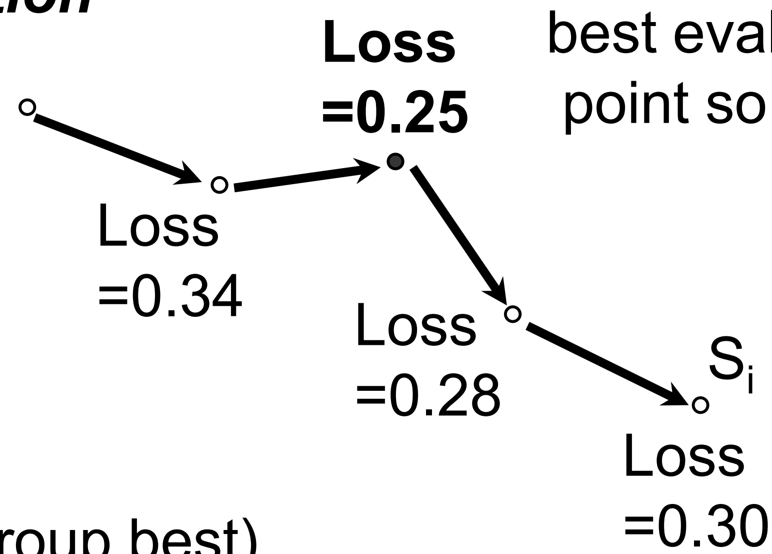
2. Basic Particle Swarm Optimization

Gbest model

Pbest and Gbest concept

During search procedure, each search point is evaluated using the ***objective function***

Pbest (personal best)
best evaluated point so far



Agent	Pbest
#1	***0.25***
#2	0.28
:	

Gbest (group best)
The best of pbests of all agents

Q: How does each agent move in the solution space and find the optimal solution ***using search histories (Pbests and Gbest)*** ?

To reply to the question

Velocity (Directions to change search points)

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k)$$

Keep going
= *Global Search*

Converging to
Pbest
= *Local Search*

Converging to
Gbest
= *Local Search*

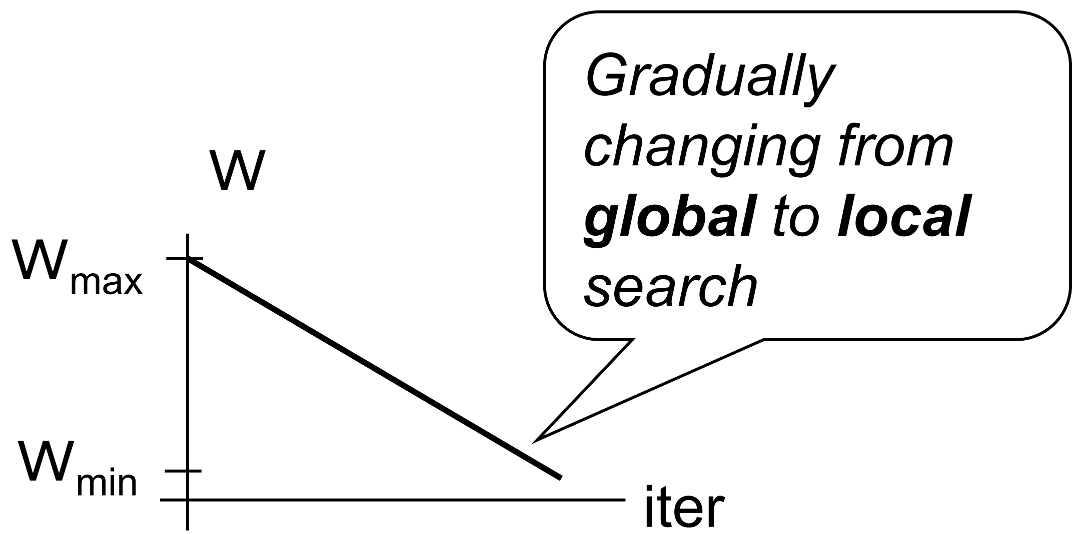
Combination of *global* and *local* search

Velocity (Directions to change searching points)

$$v_i^{k+1} = w v_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k)$$

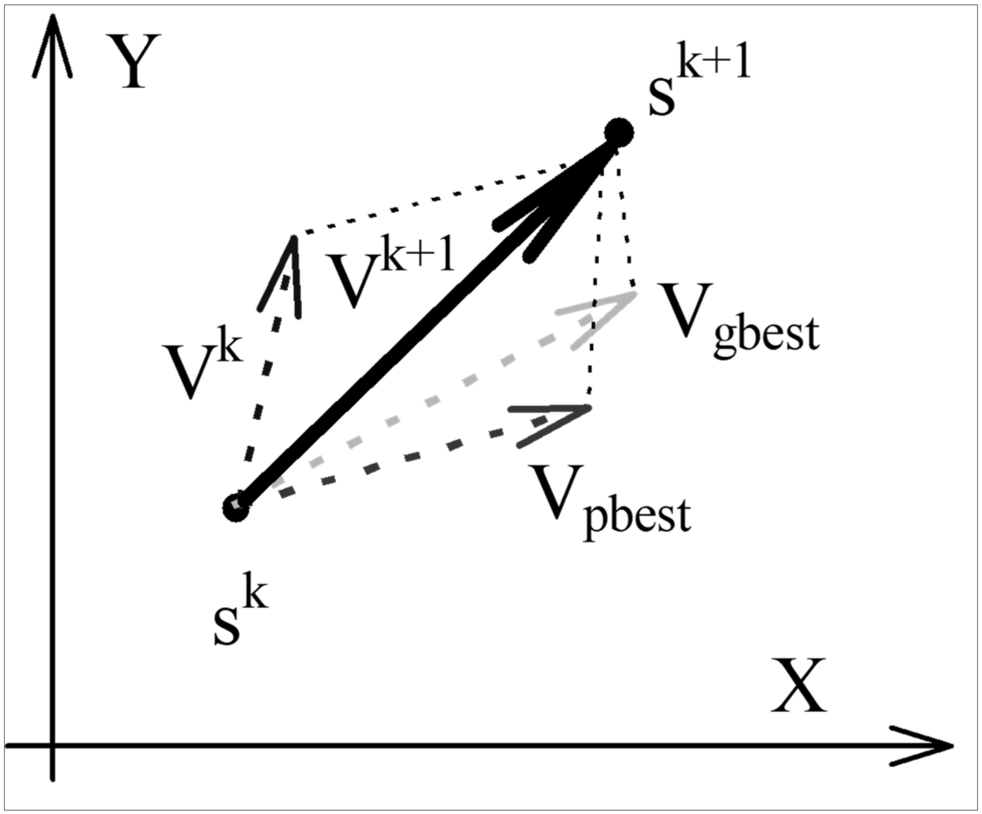
Inertia Weights Approach

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter$$

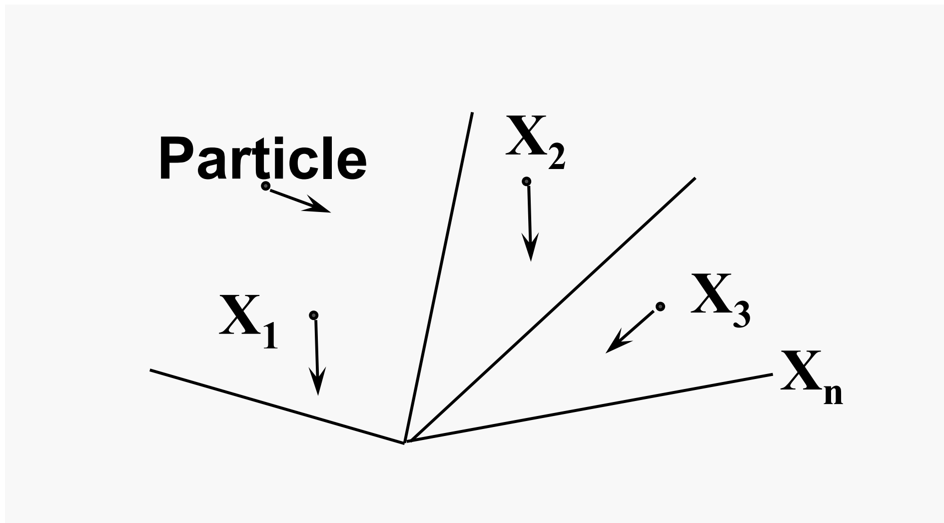


Search point

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

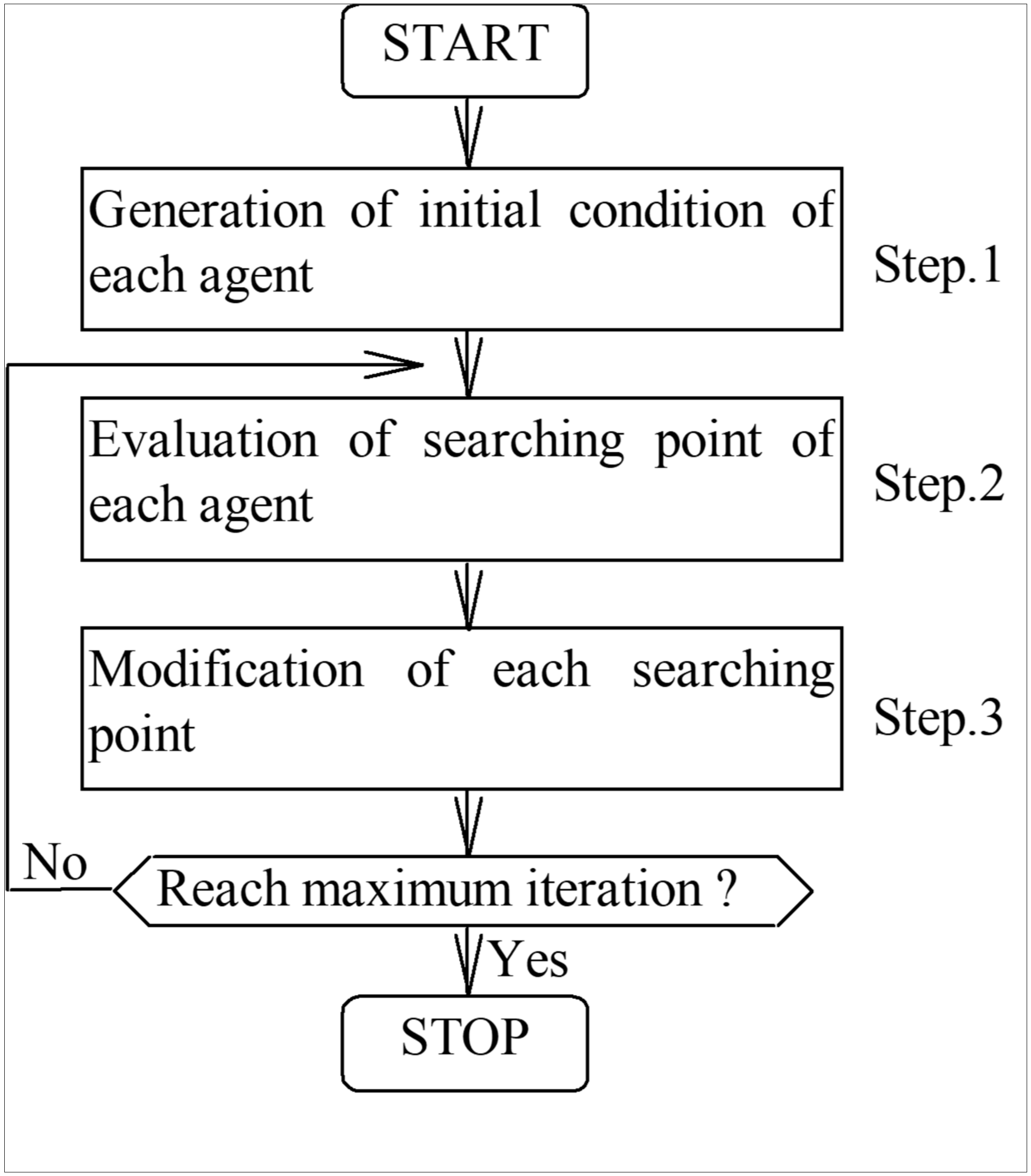


Search direction can be calculated by ***combination of vectors***



Particles (Agents) are moving as a ***Swarm*** in solution space and find an optimal solution by changing information.

A general flow chart of PSO



3. Variations of Particle Swarm Optimization

1. Discrete PSO

For Combinatorial Optimization problem

S_i = 0 or 1;   ex.) S_i = (0,1,1,1,1,0)

$$v_i^{k+1} = v_i^k + rand \times (pbest_i - s_i^k) + rand \times (gbest - s_i^k) \quad (6)$$

$$\rho_i^{k+1} < sig(v_i^{k+1}) \quad then\ s_i^{k+1} = 1; \quad (7)$$

$$else\ s_i^{k+1} = 0$$

Random number

$$sig(v_i^k) = \frac{1}{1 + \exp(-v_i^k)} \quad (5)$$

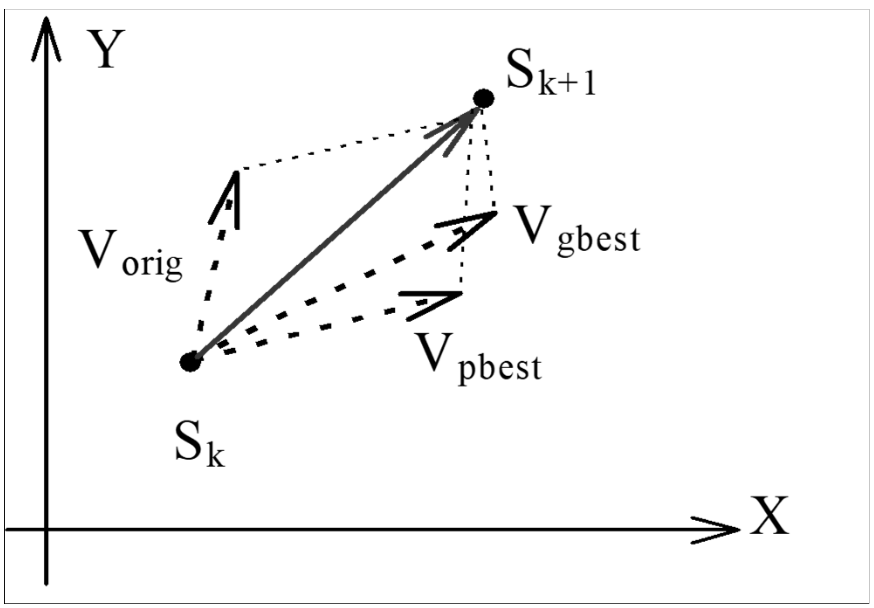
Sigmoidal Function

2. PSO for MINLP

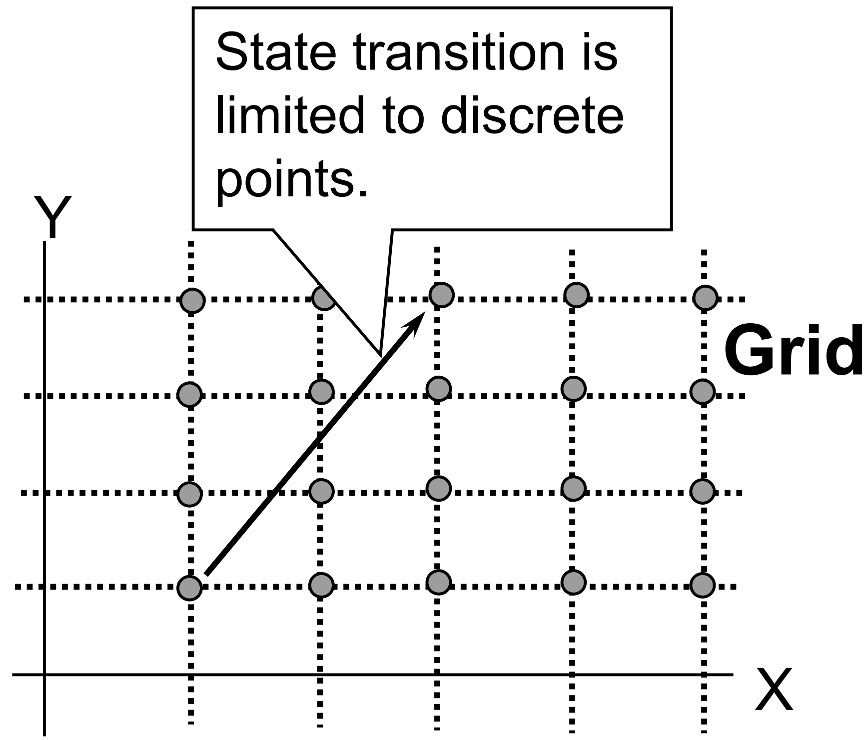
Practical problems sometimes require to handle both ***discrete*** and ***continuous*** variables.

➡ Mixed-integer Nonlinear Optimization problem (MINLP)

Continuous **Variable**



Expansion



Basic PSO

PSO for MINLP

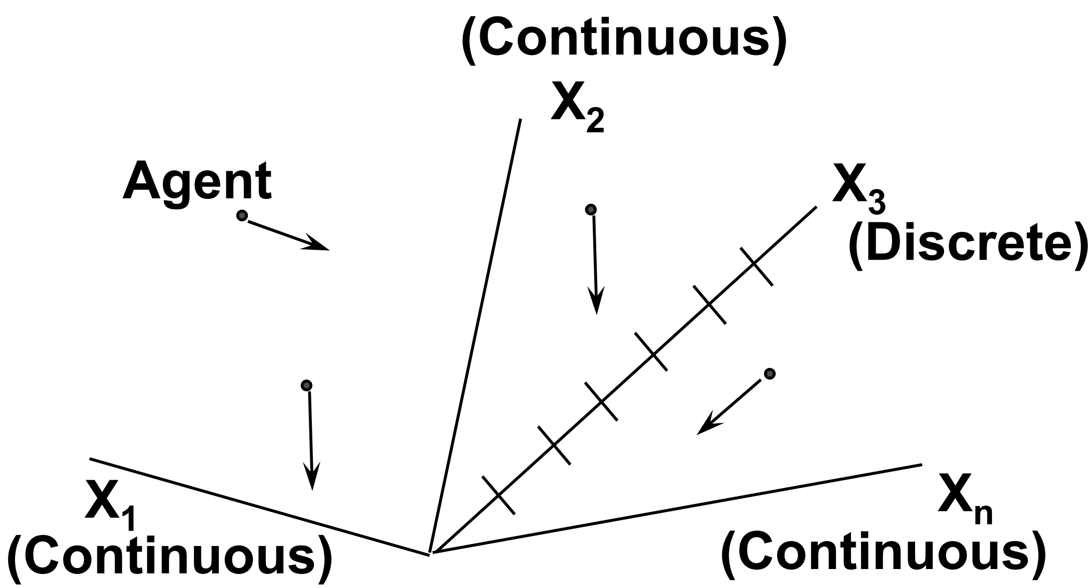
For Discrete variables

$$v_i^{k+1} = wv_i^k + c_1\underline{rand}_1 \times (pbest_i - s_i^k) + c_2\underline{rand}_2 \times (gbest - s_i^k)$$

Discretized random number *is utilized in order to put the value to the existing points in the grid*

$$\underline{s_i}^{k+1} = s_i^k + v_i^{k+1} \qquad (3)$$

Only discretized value *is allowed.*



*** both continuous and discrete variables**
*** n-dimensional problem**

Good Tool

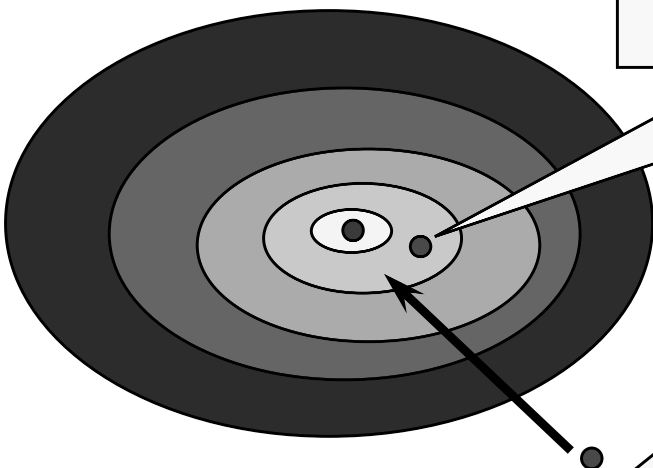
Mixed-Integer Nonlinear Optimization Problem (MINLP)

3. Hybrid PSO (HPSO)

HPSO = **PSO** + **Concept of Selection**

Highly qualified values can be found near the highly qualified values.

Knowledge

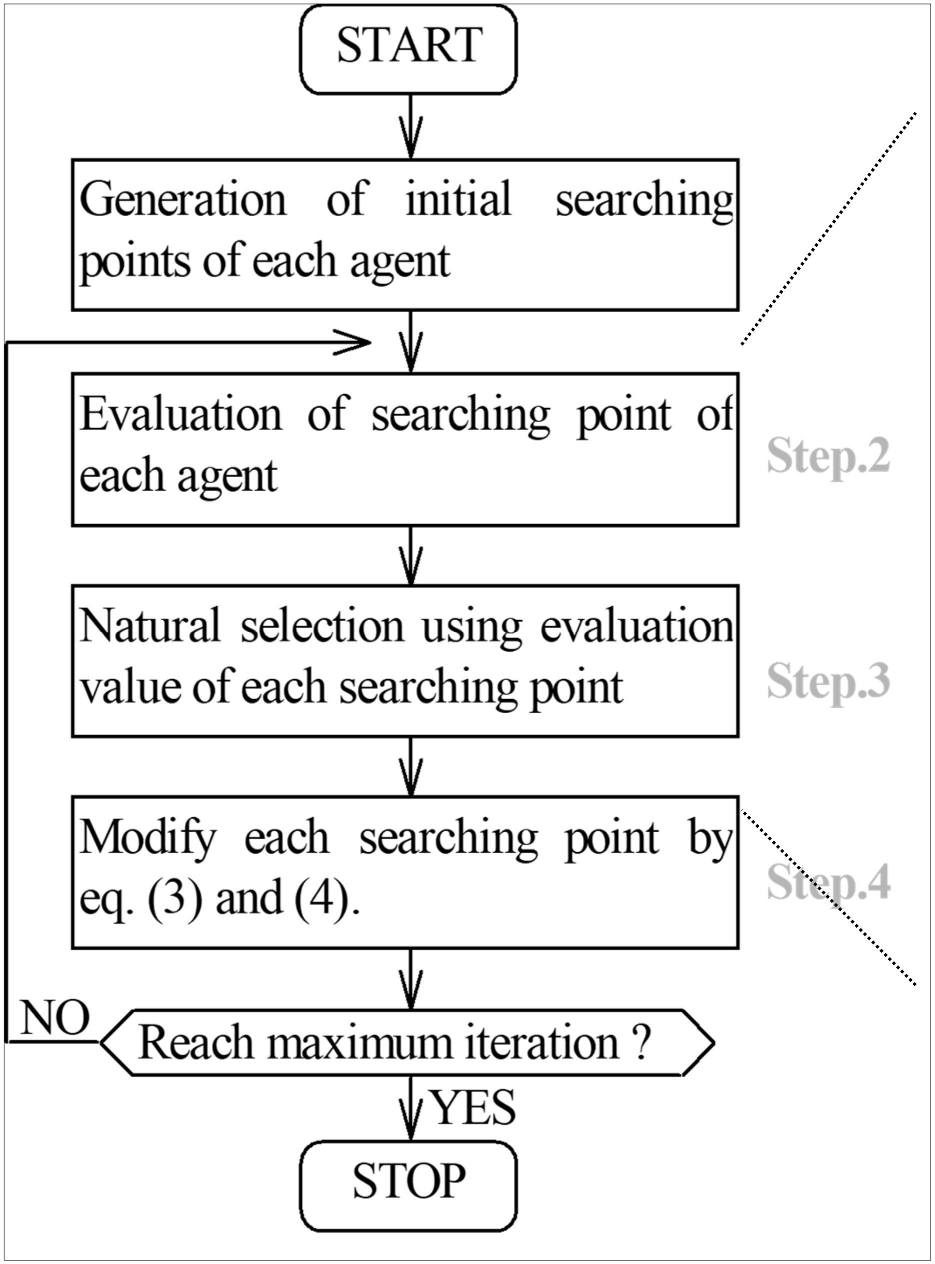


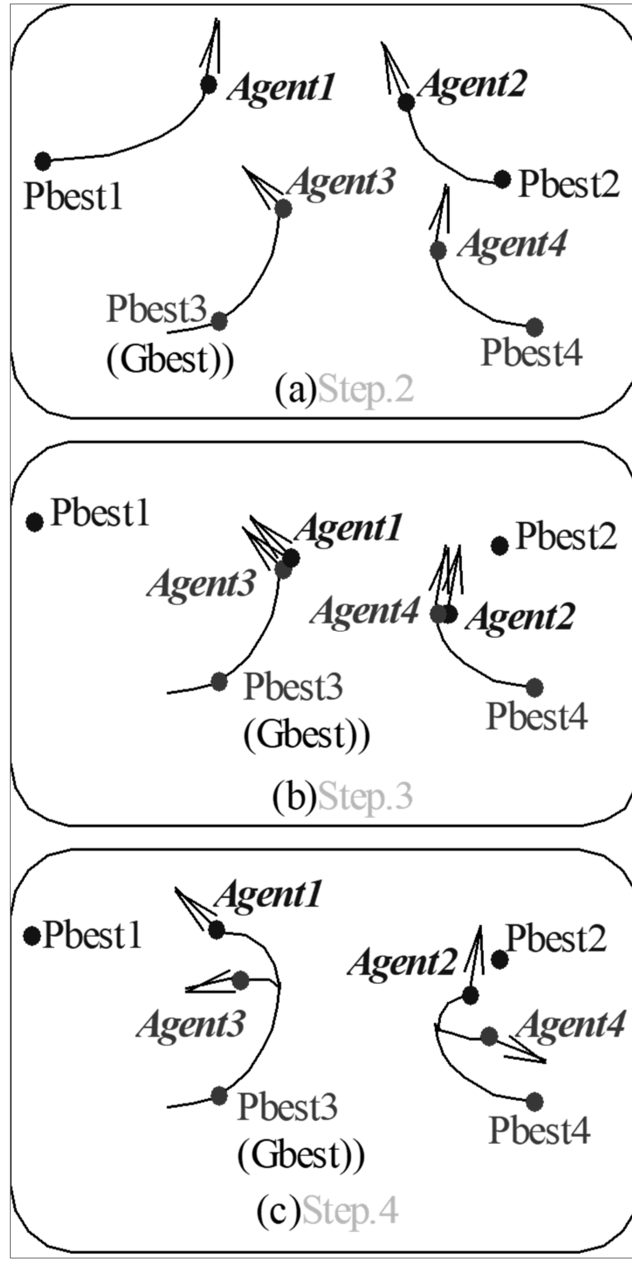
+

Better to move the *not*-highly qualified points to the points near the highly qualified points.

Introduction of ***concept of selection*** into PSO

Concept of Hybrid PSO





Evaluation values of Agent 1, 2 are low and those of agent 3, 4 are high.

Agent 1, 2 are disappeared and agent 3, 4 are double-copied by the selection mechanism. But, Pbest1 and 2 are not changed.

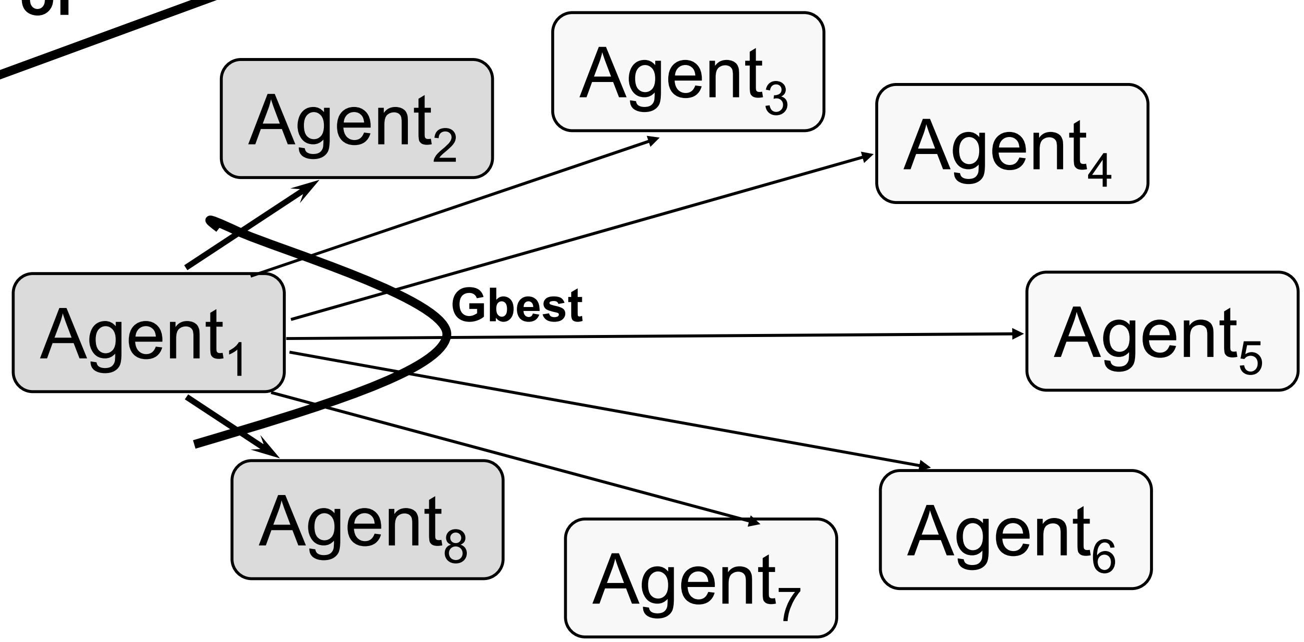
New search is begun from the new searching points using the different Pbests.

4. Lbest Model

$$v_i^{k+1} = wv_i^k + c_1 rand \times (pbest_i - s_i^k) + c_2 rand \times (lbest_i - s_i^k)$$

Getting information from **the limited number of neighbors**

$Lbest_1$ = Best{$Pbest_1$, $Pbest_2$, $Pbest_8$}

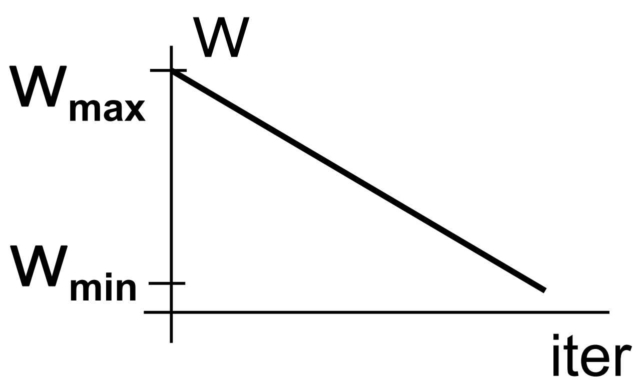


Kind of Co-evolution model

4. Parameter Selections and Constriction Factor Approach

1. Inertia Weights

$$v_i^{k+1} = w_i v_i^k + rand \times c_1(pbest_i - s_i^k) + rand \times c_2(gbest - s_i^k) \quad (2)$$



W_{max}: initial weight of the weight function
W_{min}: final weight of the weight function
C_i: weighting factor

The following parameters are appropriate and the values do ***not*** depend on problems:

w_{max}=0.9, w_{min}=0.4, c_i=2.0

2. Constriction factor approach

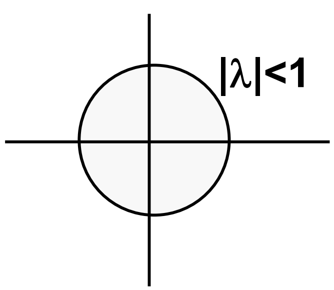
Try to control the search procedure of PSO

Search equation can be simplified by the following eqns:

$$p_{id} \leftarrow \frac{\varphi_1 p_{id} + \varphi_2 \, p_{gd}}{\varphi_1 + \varphi_2}$$

$$v_{id}(t+1) = v_{id}(t) + \varphi\left(p_{id} - x_{id}(t)\right)$$

A kind of linear discrete equation



Dynamic behavior of the system can be analyzed by **eigenvalue analysis**

2. Constriction factor approach

$$v_i^{k+1} = K[v_i^k + c_1 \times rand() \times (pbest_i - s_i^k) + c_2 \times rand() \times (gbest - s_i^k)]$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \ where\ \varphi = c_1 + c_2, \varphi > 4$$

Using the above equations, the search procedure

- **can *not* diverge in real number**
- **can search the different region efficiently**

5. Research Areas and Applications

Table 1 PSO applications.

Application field	No.
Neural network learning algorithm	[14][56]
Human tremor analysis	[7]
Rule Extraction in Fuzzy Neural Network	[33]
Battery Pack State-of-Charge Estimation	[46]
Computer Numerically Controlled Milling Optimization	[65]
Reactive Power and Voltage Control	[9][32][66]
Distribution state estimation	[50]
Power System Stabilizer Design	[16]
Fault State Power Supply Reliability Enhancement	[51]

*) No. shows the paper No. shown in ref. section.

6. Conclusions

- While many evolutionary computation techniques have been developed for combinatorial optimization problems, PSO has been basically developed for ***continuous optimization problem***.
- PSO has several variations including integration with *selection mechanism* (HPSO) and hybridization for handling both *discrete and continuous* variables (PSO for MINLP).
- ***Constriction factor approach*** is based on mathematical analysis and can be useful for obtaining high quality solutions.
- PSO can be an efficient optimization tool for ***nonlinear continuous optimization problems, combinatorial optimization problems, and mixed-integer nonlinear optimization problem (MINLP)***.