SCS5107 Computational Intelligence Unit III

Particle Swarm Optimization (PSO)

PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms.PSO applies the concept of social interaction to problem solving.

It was developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer). 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.

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 accelaration at each time step as shown in Fig.1

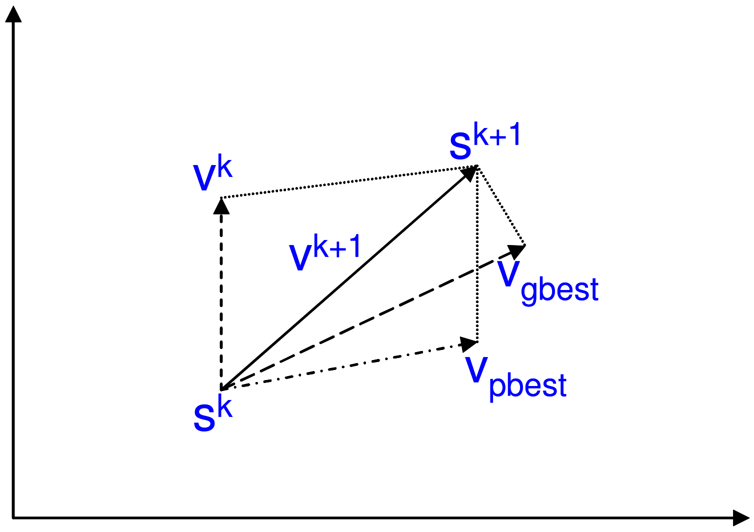


Fig.1 Concept of modification of a searching point by PSO

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

- 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.

- The modification of the particle's position can be mathematically modeled according the following equation :

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

where, 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.

The following weighting function is usually utilized in (1)

$$\mathbf{\mathit{w} = wMax-[(wMax-wMin) \ x \ iter]/maxIter \qquad (2)}$$

where wMax= initial weight,

wMin = final weight,

maxIter = maximum iteration number,

iter = current iteration number.

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

Comments on the Inertial weight factor:

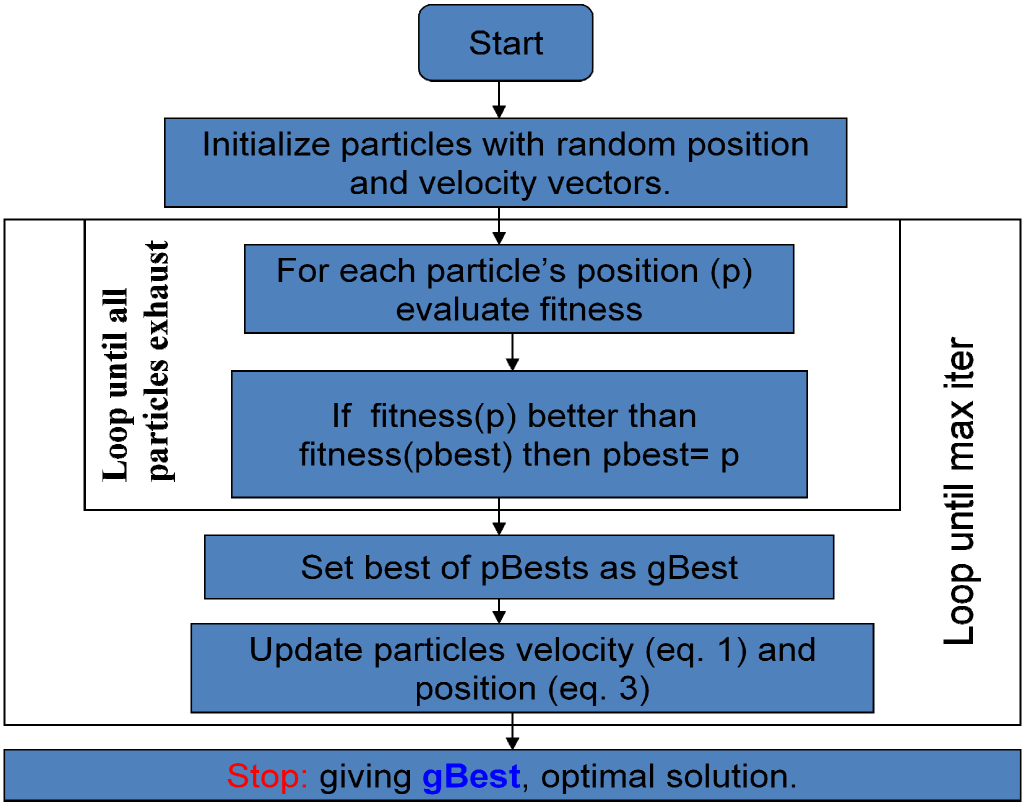
A large inertia weight (*w*) facilitates a global search while a small inertia weight facilitates a local search.

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.

Flow chart depicting the General PSO Algorithm:


Start
Initialize particles with random position and velocity vectors.
Loop until all particles exhaust
For each particle's position (p) evaluate fitness
If fitness(p) better than fitness(pbest) then pbest= p
Loop until max iter
Set best of pBests as gBest
Update particles velocity (eq. 1) and position (eq. 3)
Stop: giving gBest, optimal solution.

PSO Algorithm

```
[x*] = PSO()
$P$ = Particle_Initialization();
For $i$=1 to $it\_max$
  For each particle $p$ in $P$ do
    $fp$ = f($p$);
    If $fp$ is better than f($pBest$)
        $pBest = p$;
    end
  end
  $gBest$ = best $p$ in $P$;
  For each particle $p$ in $P$ do
     $v = v + c1*rand*(pBest - p) + c2*rand*(gBest - p)$;
     $p = p + v$;
  end
end
```

1. Create a ‘population’ of agents (particles) uniformly distributed over X

2. Evaluate each particle’s position according to the objective function

3. If a particle’s current position is better than its previous best position, update it

4. Determine the best particle (according to the particle’s previous best positions)

5. Update Particles Velocity

$$\boxed{\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^{t}}_{inertia} + \underbrace{c_1 \mathbf{U}_1^{t}(\mathbf{pb}_i^{t} - \mathbf{p}_i^{t})}_{personal\ influence} + \underbrace{c_2 \mathbf{U}_2^{t}(\mathbf{gb}^{t} - \mathbf{p}_i^{t})}_{social\ influence}}$$

6.Move particles to the new position

$$\boxed{\mathbf{p}_i^{t+1} = \mathbf{p}_i^{t} + \mathbf{v}_i^{t+1}}$$

7. Go to step 2 until stopping criteria is satisfied

Ant Algorithms

- Ants are practically blind but they still manage to find their way to and from food. How do they do it? These observations inspired a new type of algorithm called ***ant algorithms*** (or ant systems). These algorithms are very new (Dorigo, 1996) and is still very much a research area. Ant systems are a population based approach. In this respect it is similar to genetic algorithms

There is a population of ants, with each ant finding a solution and then communicating with the other ants.

We are interested in exploring the search space, rather than simply plotting a route

We need to allow the ants to explore paths and follow the best paths with some *probability* in proportion to the intensity of the pheromone trail

We do not want them simply to follow the route with the highest amount of pheromone on it, else our search will quickly settle on a sub-optimal (and probably very sub-optimal) solution.

Stigmergy is:

indirect communication via interaction with the environment [Gassé, 59]

- A problem gets solved bit by bit ..
- Individuals communicate with each other in the above way, affecting what each other does on the task.

E.g. sorting things into piles, as we did in the Introductory Swarm Intelligence lecture

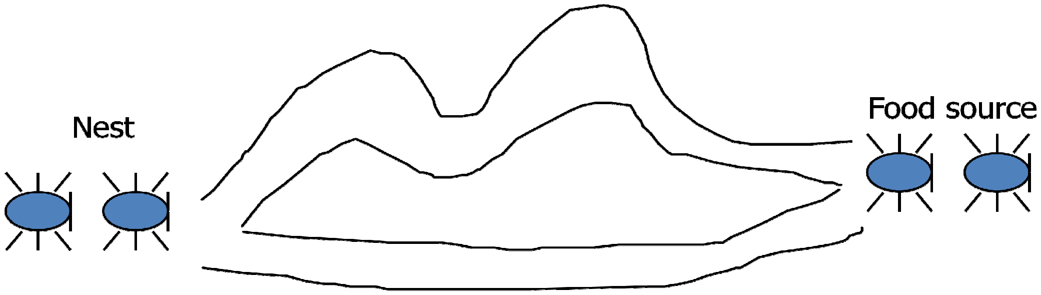
- Individuals leave *markers* or *messages* – these don't solve the problem in themselves, but they affect other individuals in a way that helps them solve the problem …
- E.g. as we will see, this is how ants find shortest paths.

Pheromone Trails

Individual ants lay pheromone trails while travelling from the nest, to the nest or possibly in both directions.

The pheromone trail gradually evaporates over time.

But pheromone trail strength accumulate with multiple ants using path.



Ant Colony Optimisation Algorithms: Basic Ideas

Ants are *agents* that:

Move along between nodes in a graph.They choose where to go based on pheromone strength (and maybe other things).An ant's path represents a specific candidate solution.When an ant has finished a solution, pheromone is laid on its path, according to quality of solution.This pheromone trail affects behaviour of other ants by `stigmergy' …

The transition rule

T(*r*,s) is the amount of pheromone currently on the path that goes directly from city *r* to city *s*.

H(*r*,*s*) is the heuristic value of this link – in the classic TSP application, this is chosen to be 1/distance(*r*,*s*) -- I.e. the shorter the distance, the higher the heuristic value.

$p_k(r,s)$ is the probability that ant *k* will choose the link that goes from *r* to *s*

β is a parameter that we can call the *heuristic strength*

$$p_k(r,s) = \frac{T(r,s) \cdot H(r,s)^{\beta}}{\sum_{\text{unvisited cities } c} T(r,c) \cdot H(r,c)^{\beta}}$$

- Elitist strategy

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} e / L^{gb}(t) & \text{if } \operatorname{arc}(i,\ j) \in T^{gb} \\ 0 & \text{otherwise} \end{cases}$$

- AS_{rank}

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{r=1}^{w-1}(w-r)\Delta\tau_{ij}^{r}(t) + w\Delta\tau_{ij}^{gb}(t)$$

- ACS
  - Strong elitist strategy
  - Pseudo-random proportional rule

$$j = \arg\ \max_{j \in N_i^k} \left\{ \tau_{ij}(t)\eta_{ij}^{\beta} \right\}$$

With Probability (1- q_0):

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & \text{if } j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases}$$

ACS (Pheromone update)

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^{best}(t)$$

Artificial Bee Colony

Bonabeau has defined swarm intelligence as

"any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies" [1]

- Term swarm in general refer to any restrained collection of interacting agents or individuals

- Two fundamental concepts self-organization and division of labour, are necessary and sufficient properties to obtain swarm intelligent behaviour

- Self-organization : can be defined as a set of dynamical mechanisms that establish basic rules for interactions between the components of the system. The rules ensure that the interactions are executed on the basis of purely local information without any relation to the global pattern.

- Bonabeau et al. has characterized four basic properties on which self organization relies:

  - Positive feedback

  - Negative feedback,

  - Fluctuations and

  - Multiple interactions

- Division of Labor: In swarm behavior different tasks are performed simultaneously by specialized individuals which is referred to as division of labor. It enables swarm to respond to changed conditions in the search space.

Behavior of Honey Bee Swarm

Three essential components of forage selection:

- Food Sources: The value of a food source depends on many factors such as its proximity to the nest, its richness or concentration of its energy, and the ease of extracting this energy.

- Employed Foragers: They are associated with a particular food source which they are currently exploiting or are "employed" at. They carry with them information about this particular source, its distance and direction from the nest, the profitability of the source and share this information with a certain probability.

- Unemployed Foragers: They are continually at look out for a food source to exploit. There are two types of unemployed foragers: scouts, searching the environment surrounding the nest for new food sources and onlookers waiting in the nest and establishing a food source through the information shared by employed foragers.
- The model defines two leading modes of the behavior:
  - recruitment to a nectar source
  - the abandonment of a source.

Exchange of Information among bees:

- The exchange of information among bees is the most important occurrence in the formation of collective knowledge.
- The most important part of the hive with respect to exchanging information is the dancing area
- Communication among bees related to the quality of food sources takes place in the dancing area.
- This dance is called a Waggle dance.
- Employed foragers share their information with a probability proportional to the profitability of the food source, and the sharing of this information through waggle dancing is longer in duration.
- An onlooker on the dance floor, probably she can watch numerous dances and decides to employ herself at the most profitable source.
- There is a greater probability of onlookers choosing more profitable sources since more information is circulated about the more profitable sources.

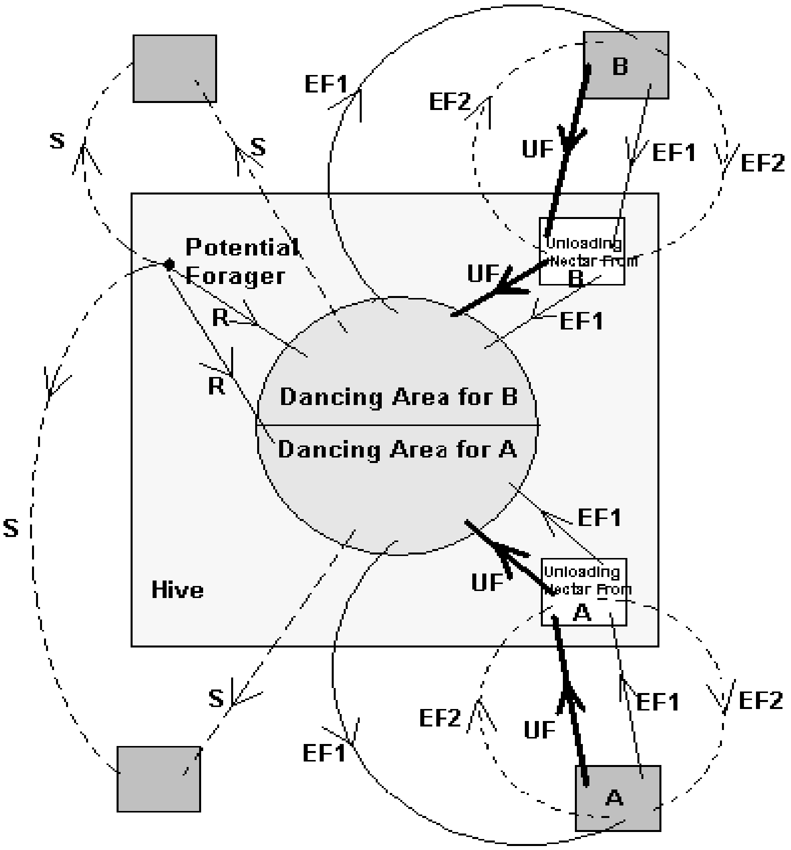
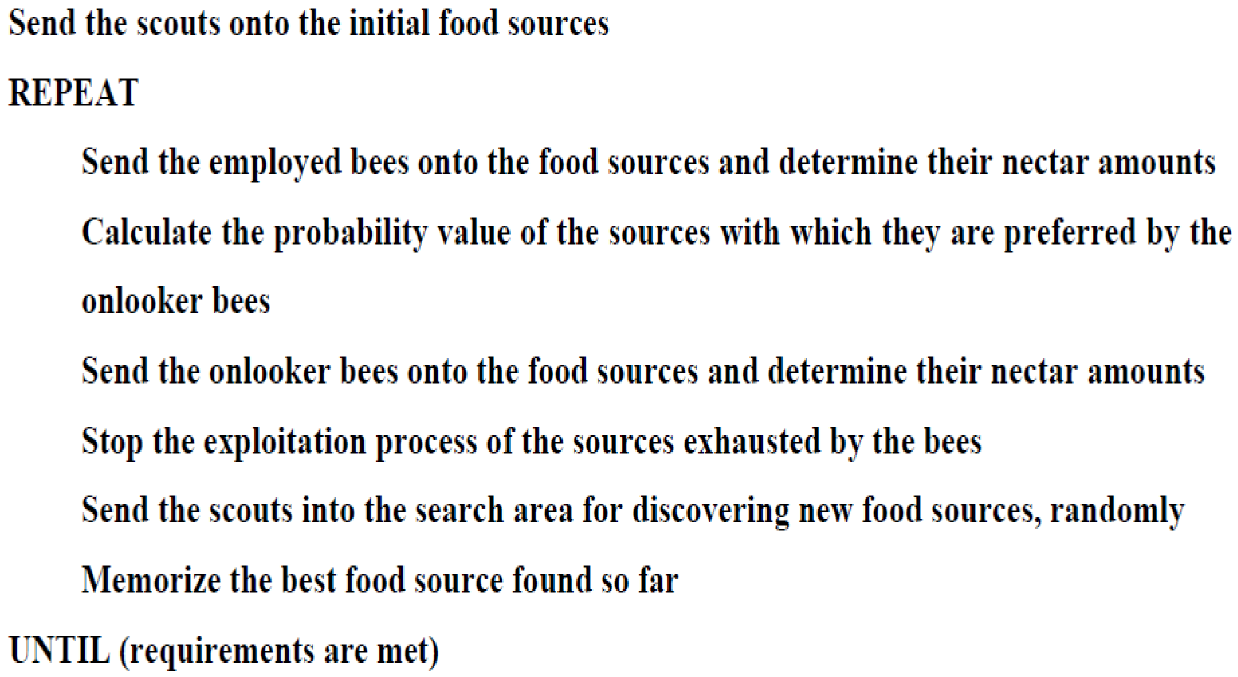


Figure 1. The behaviour of honey bee foraging for nectar

Basic Self Organization Properties

- Positive feedback: As the nectar amount of food sources increases, the number of onlookers visiting them increases, too.
- Negative feedback: The exploitation process of poor food sources is stopped by bees.
- Fluctuations: The scouts carry out a random search process for discovering new food sources.
- Multiple interactions: Bees share their information about food sources with their nest mates on the dance area.

ABC Algorithm



```
Send the scouts onto the initial food sources
REPEAT
    Send the employed bees onto the food sources and determine their nectar amounts
    Calculate the probability value of the sources with which they are preferred by the
    onlooker bees
    Send the onlooker bees onto the food sources and determine their nectar amounts
    Stop the exploitation process of the sources exhausted by the bees
    Send the scouts into the search area for discovering new food sources, randomly
    Memorize the best food source found so far
UNTIL (requirements are met)
```

- Each cycle of search consists of three steps: moving the employed and onlooker bees onto the food sources and calculating their nectar amounts; and determining the scout bees and directing them onto possible food sources.
- A food source position represents a possible solution to the problem to be optimized.
- The amount of nectar of a food source corresponds to the quality of the solution
- Onlookers are placed on the food sources by using a probability based selection process.
- As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases, too.
- The scouts are characterized by low search costs and a low average in food source quality. One bee is selected as the scout bee.
- The selection is controlled by a control parameter called "limit".
- If a solution representing a food source is not improved by a predetermined number of trials, then that food source is abandoned and the employed bee is converted to a scout.

Control Parameters of ABC Algorithm

- swarmsize

- Limit
- number of onlookers: 50% of the swarm
- number of employed bees: 50% of the swarm
- number of scouts: 1

ABC algorithm in fact employs four different selection processes:

- A global selection process used by the artificial onlooker bees for discovering promising regions as
- A local selection process carried out in a region by the artificial employed bees and the onlookers depending on local information
- A local selection process called greedy selection process carried out by all bees in that if the nectar amount of the candidate source is better than that of the present one, thebee forgets the present one and memorizes the candidate source. Otherwise the bee keeps the present one in the memory.

A random selection process carried out by scouts.