UNIVERSITÉ BISHOP'S UNIVERSITY

Collective intelligence

(Particle Swarm Optimization)

CS417

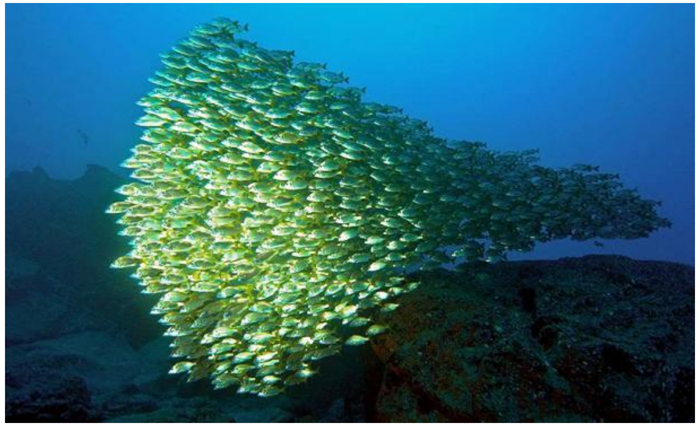
R. Hedjam

Adapted from M. Clerc

Swarming

- Why do animal swarm?

Flocking birds



Fish schooling and shoaling

Bee swarming

Flocking sheep

Collective intelligence

- **Shoaling**: a group of fish staying together,
- **Schooling**: a group of fish swimming in the same direction.
- How is this cooperation coordinated and controlled? By a master fish?

Schooling

Shoaling.

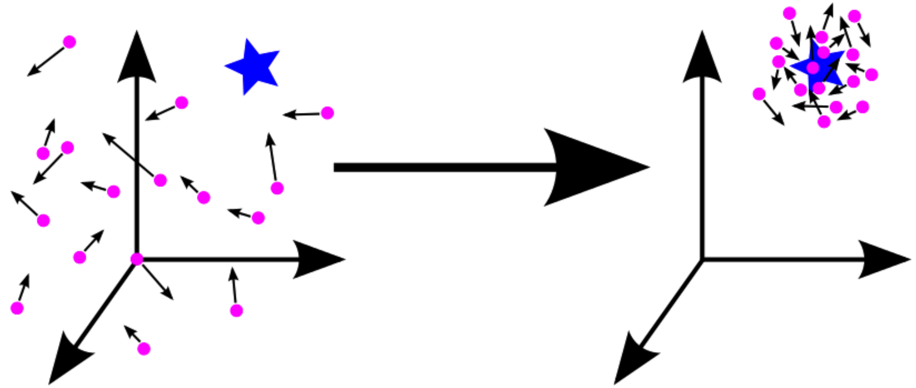
wikipedia

Collective intelligence

- Fish are:
  - shoaling → **interacting** to each other.
  - schooling → **learning** from each others.
- Swarming is an example of the emergence of a property that a group of animals possesses while the individual in the group does not.
- Swarming is a common property among many animal species.
- Collective intelligence: groups of animals work together to achieve a common goal.
- Question: How can we draw inspiration from collective intelligence to solve optimization problems?

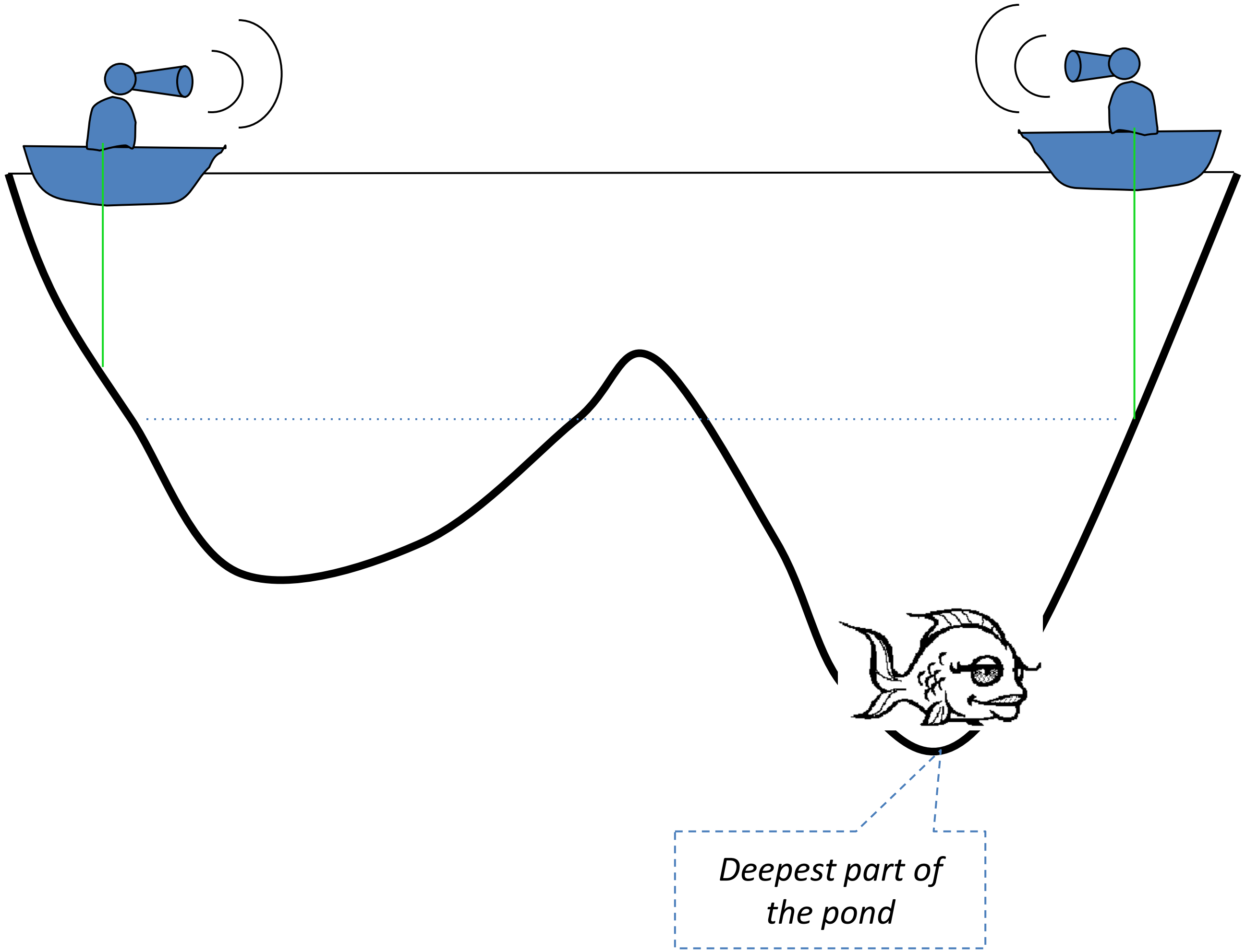
Particle swarm optimization (PSO)

- Proposed by Kennedy, Eberhart and Shi in 1995-1998.
- Originally intended to simulate social behavior, as a simplified representation of the movement of organisms in a flock of birds or a school of fish.



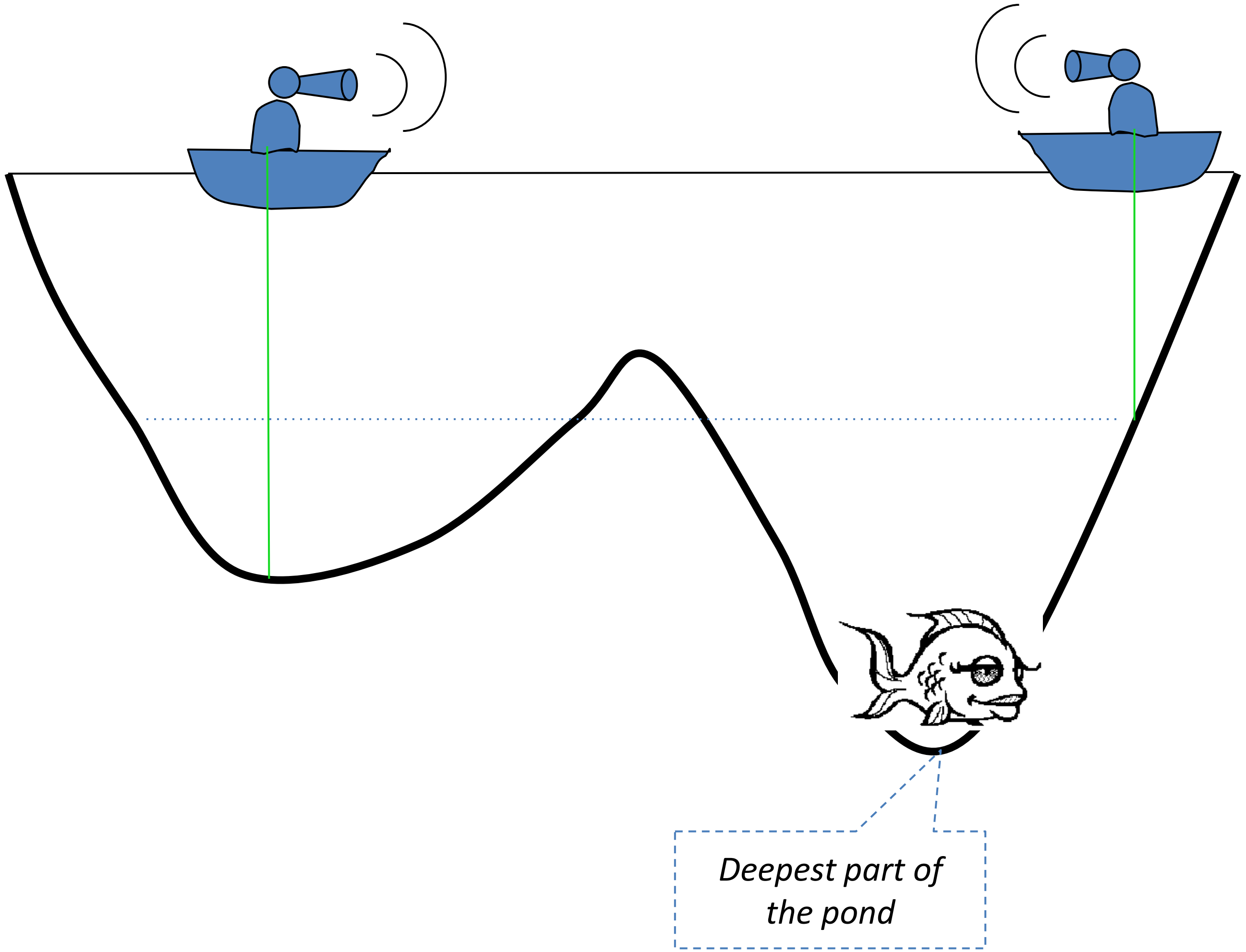
Iteration # 0   Iteration # N

PSO - Inspiration



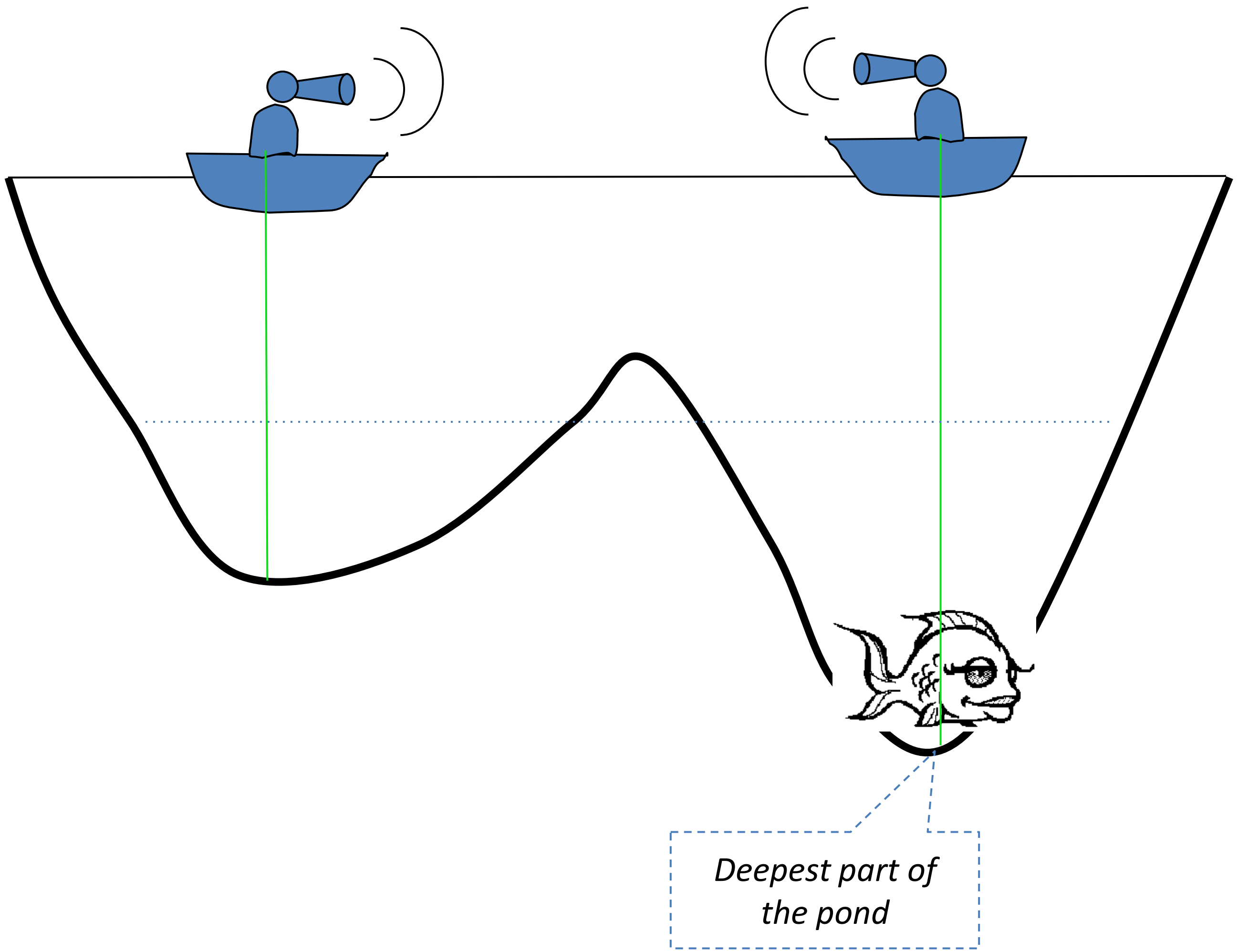
From Maurice Clerc

PSO - Inspiration



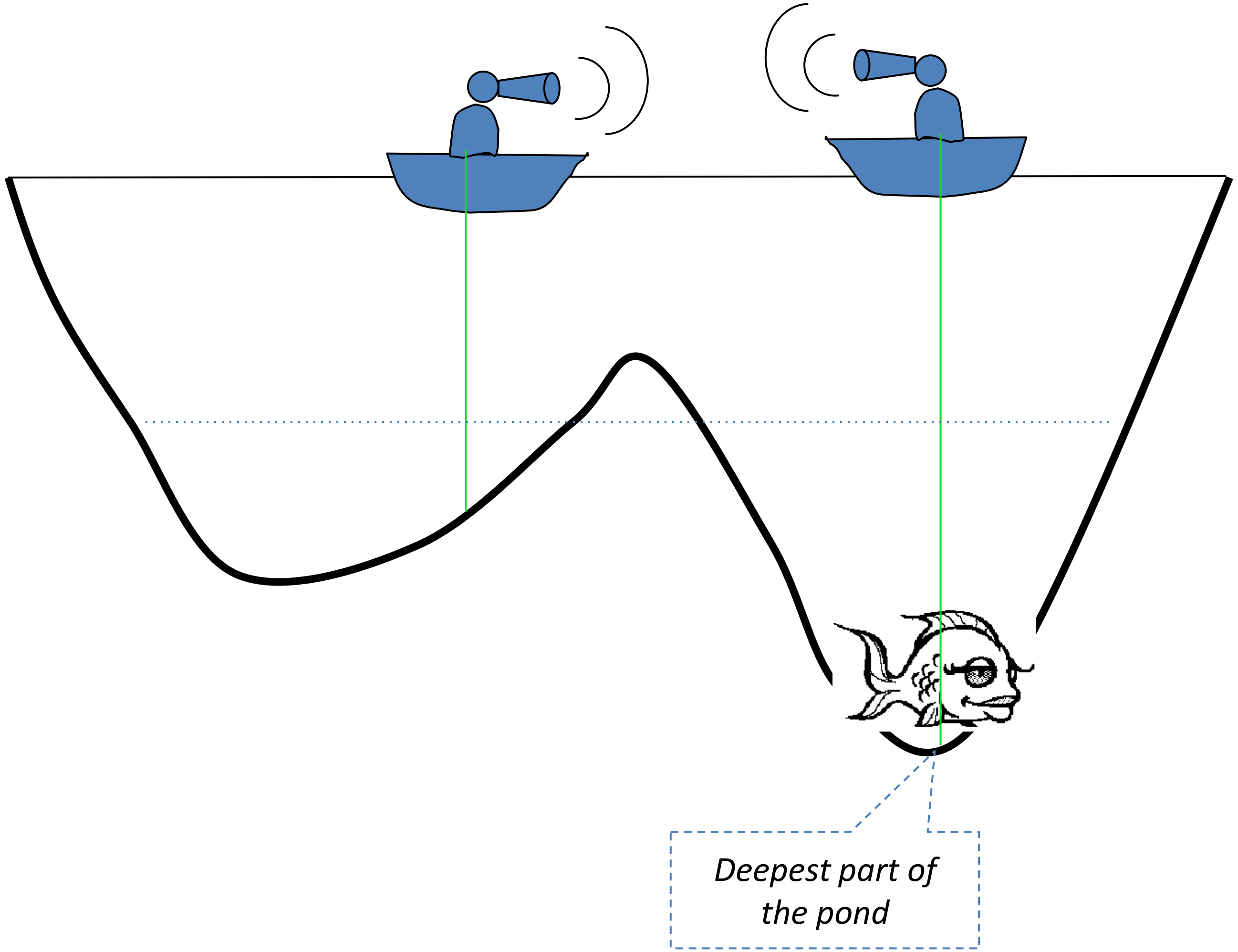
From Maurice Clerc

PSO - Inspiration



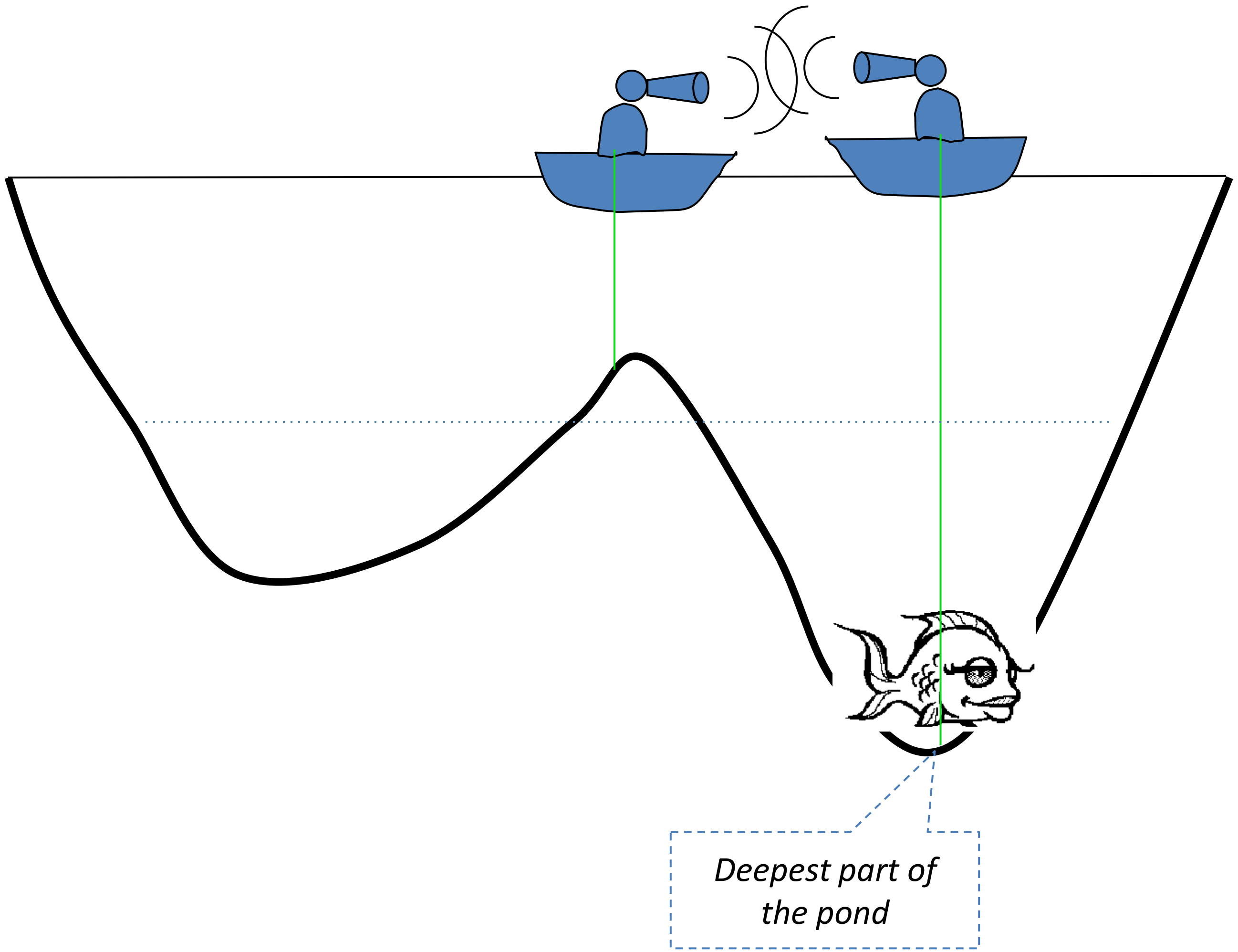
From Maurice Clerc

PSO - Inspiration



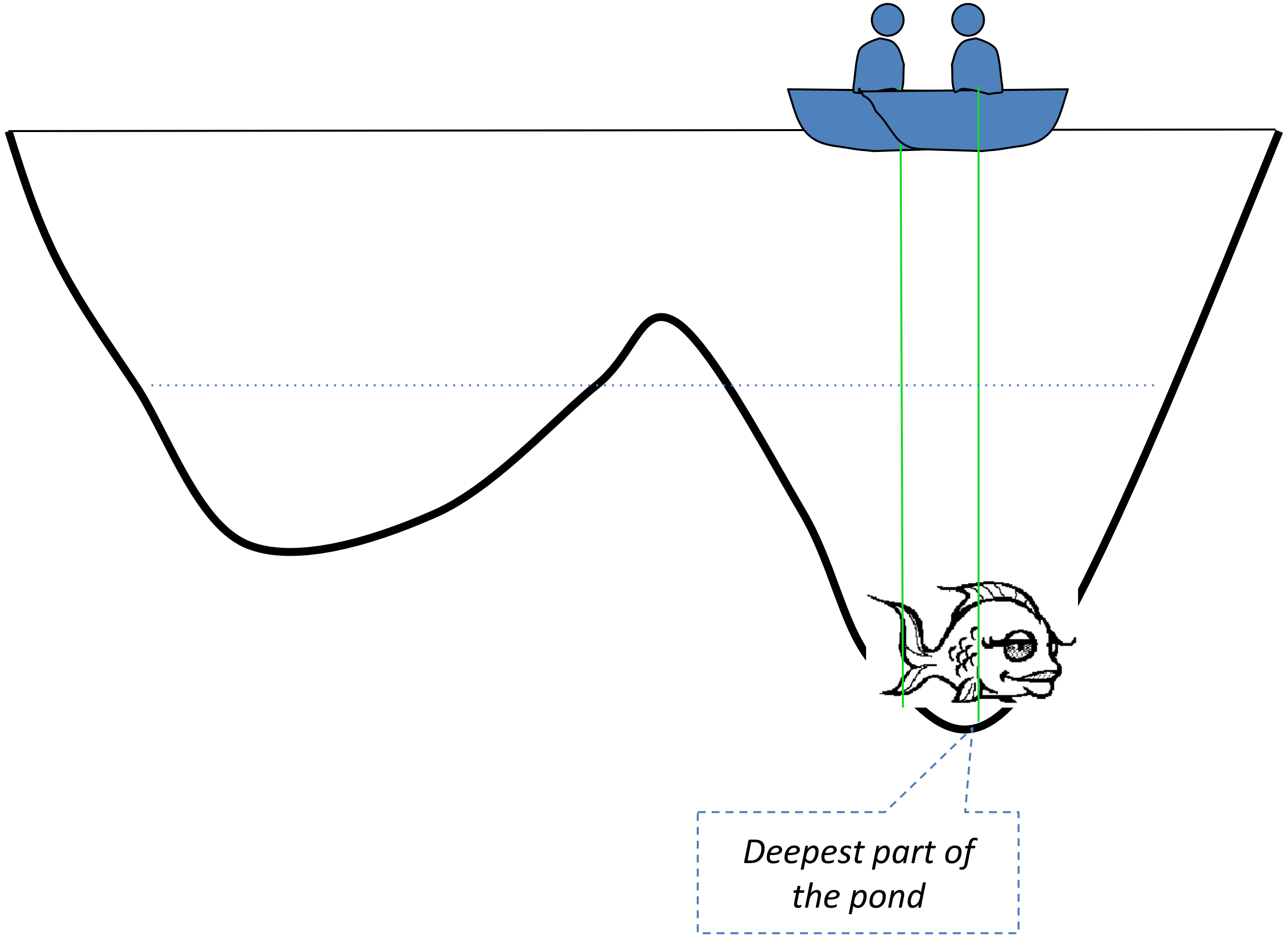
From Maurice Clerc

PSO - Inspiration



From Maurice Clerc

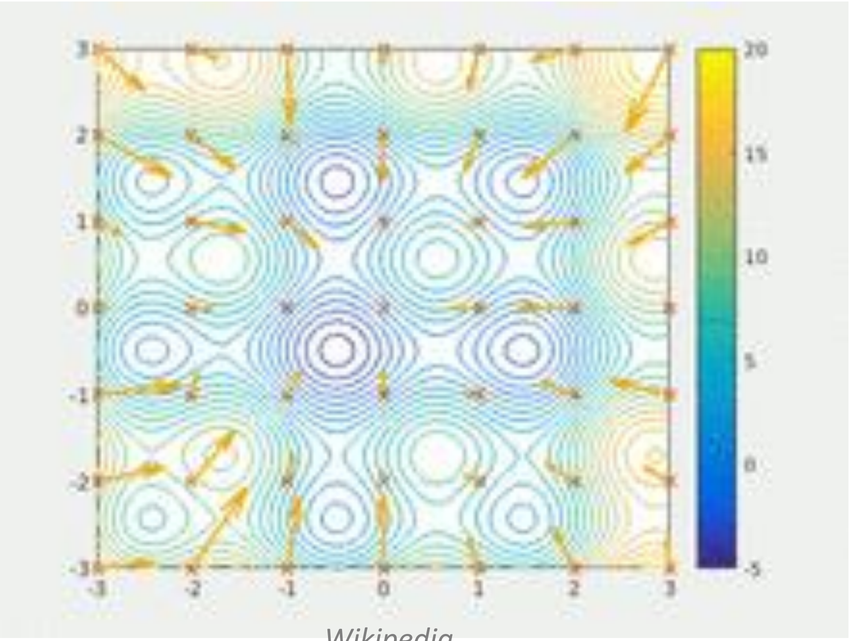
PSO - Inspiration



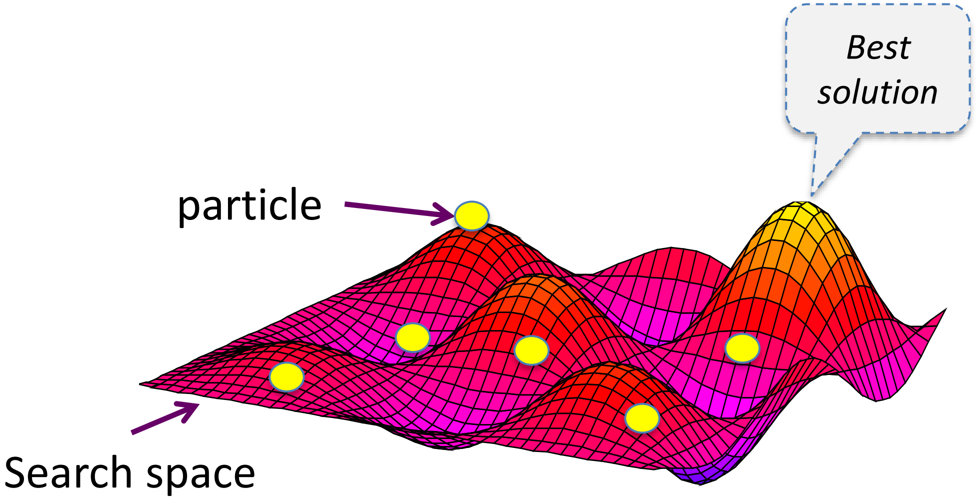
From Maurice Clerc

Particle swarm optimization (PSO)

- A computational method for optimizing a problem by iteratively improving candidate solutions (particles) based on the resulting quality measure.

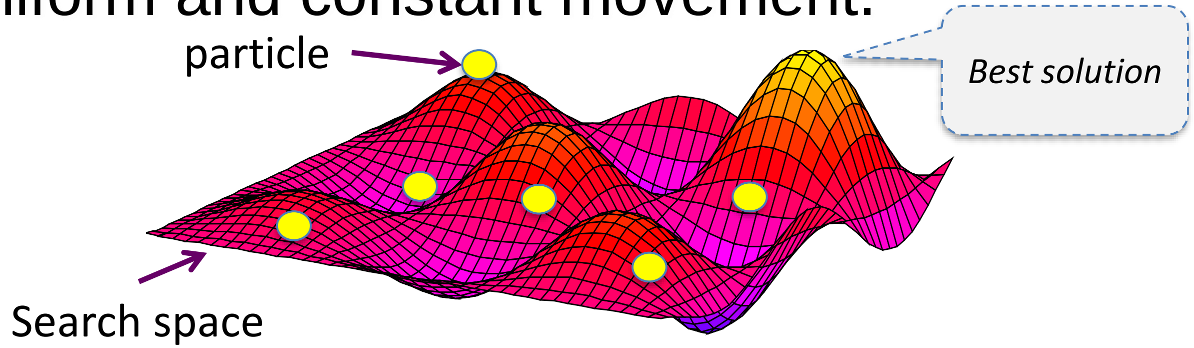


Wikipedia



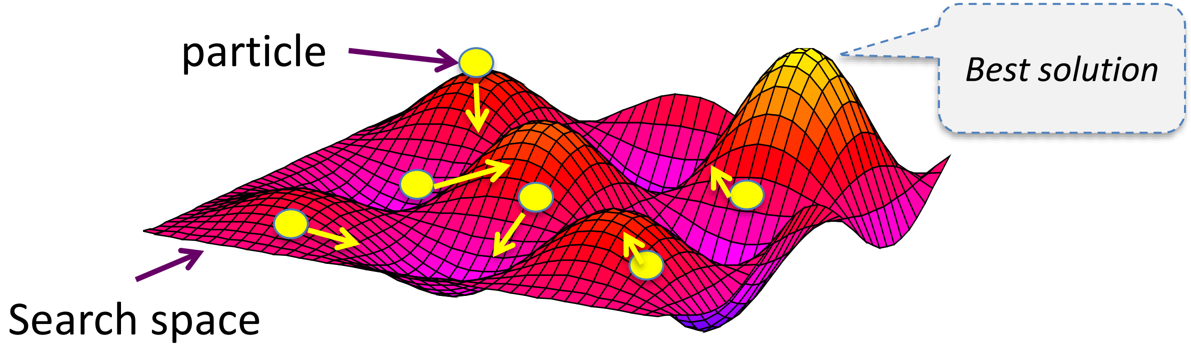
PSO search space

- PSO uses several particles that constitute a swarm in the search space looking for the best solution.
- Each particle is represented as a point (candidate solution) in a N-dimensional space which adjusts its movement according to its own movement experience as well as the movement experience of other particles.
- Particles monitor their neighbors’ movements and adapt to them to avoid collisions.
- This simple rule gives the basic uniform and constant movement.

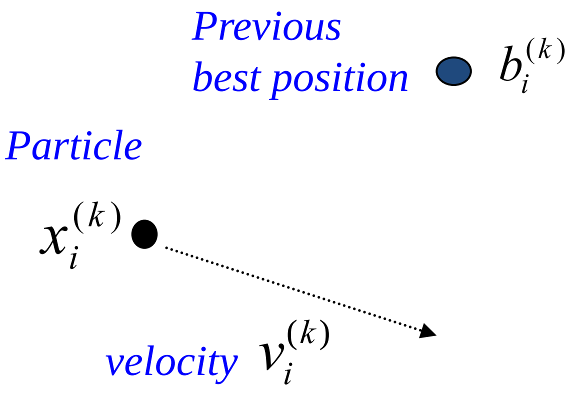


PSO process

1. Initialize a population of particles in the search space.
2. Evaluate the fitness of individual particles.
3. Modify solution based on previous best and global (or neighborhood) best solution.
4. Terminate on some condition.
5. Go to step 2.



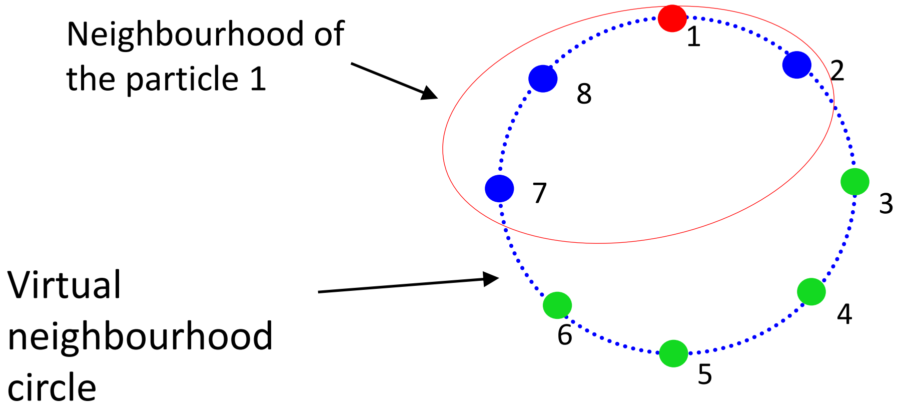
PSO - properties

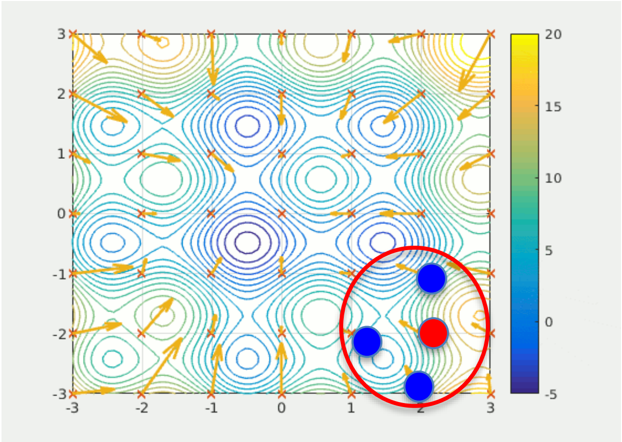


- A particle has:
  - a position,
  - a velocity (movement operator which can be applied to a position to modify it),
  - ability to exchange the information with neighbors,
  - ability to memorize the previous position,
  - ability to use information to make a decision (*help in figuring out where to search*).

PSO – properties

- The particles in the swarm *co-operate*. They exchange information about what they've discovered in the places they have visited.
- The co-operation is very simple. In basic PSO:
  - A particle has a *neighborhood* associated with it.
  - A particle knows the fitness of those in its neighborhood, and uses the *position* of the one with best fitness to adjust the particle's velocity

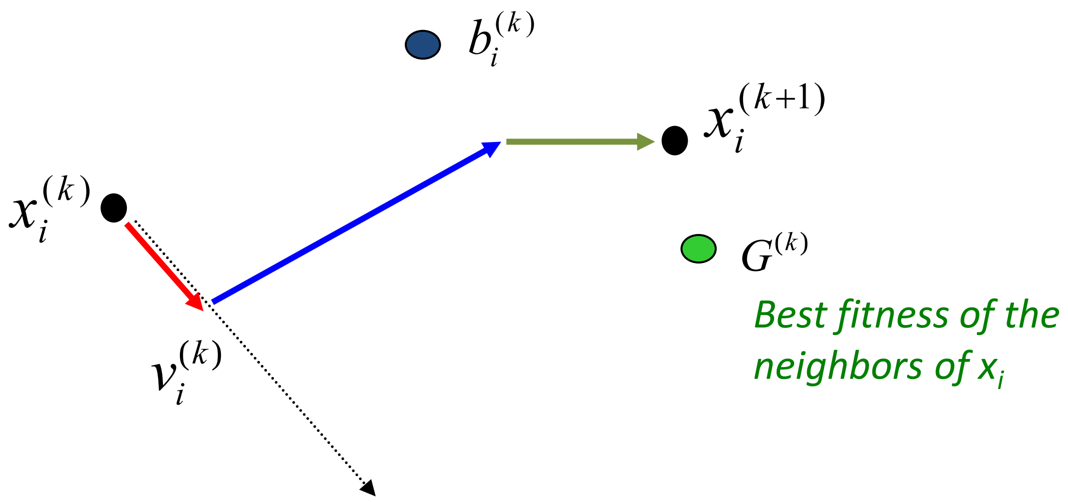




PSO – process

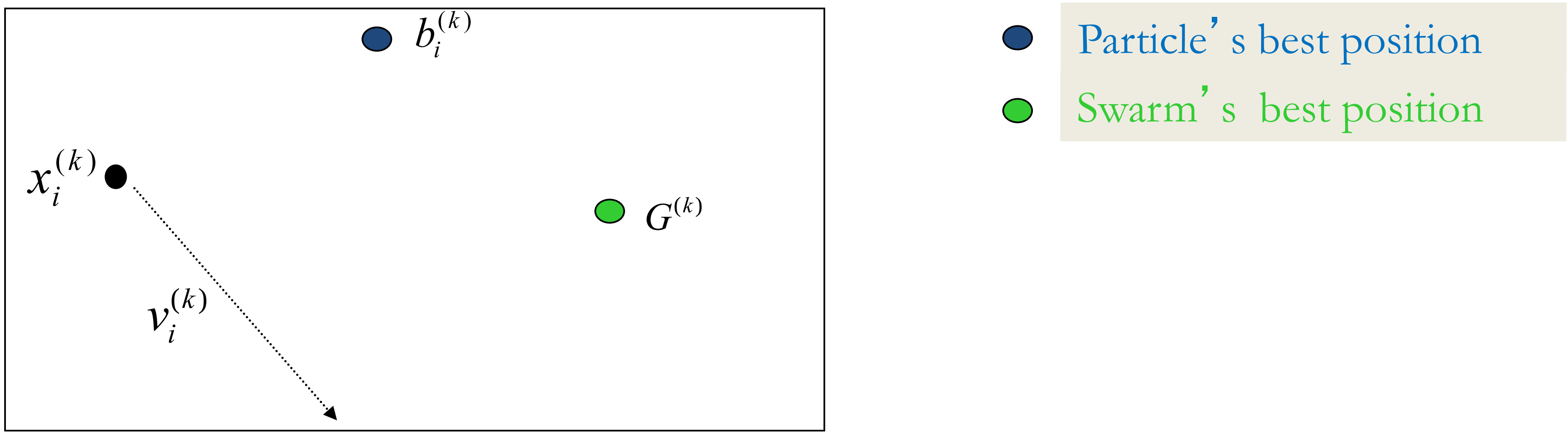
- In each iteration, a particle must move to a new position by updating its velocity:
  - New position = old position + velocity (movement's operator).
  - Velocity = a portion in the direction of its best personal position + a portion in the direction of the neighborhood best position.

$$x_i^{(k+1)} = x_i^{(k)} + v_i^{(k+1)}$$



Particles update their positions according to a: "psychological and social trade-off" between what an individual feels comfortable with and what society (swarm) values (estimates).

Note: The velocity of the particle can't exceed a maximum velocity (Vmax).

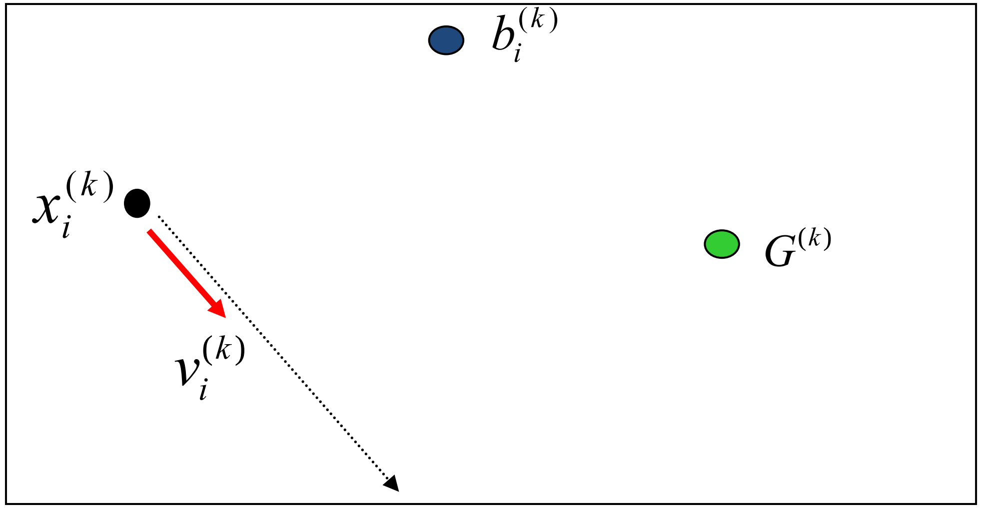


$x_i^{(k)}$ is the position of the ith particle at step k

$v_i^{(k)}$ is its velocity (movement)

$b_i^{(k)}$ is the best position visited by the ith particle

$G^{(k)}$ is the overall best position ever visited



Particle's best position

Swarm's best position

$$v_i^{(k+1)} = w.v_i^{(k)}$$

$x_i^{(k)}$ is the position of the i^{th} particle at step k

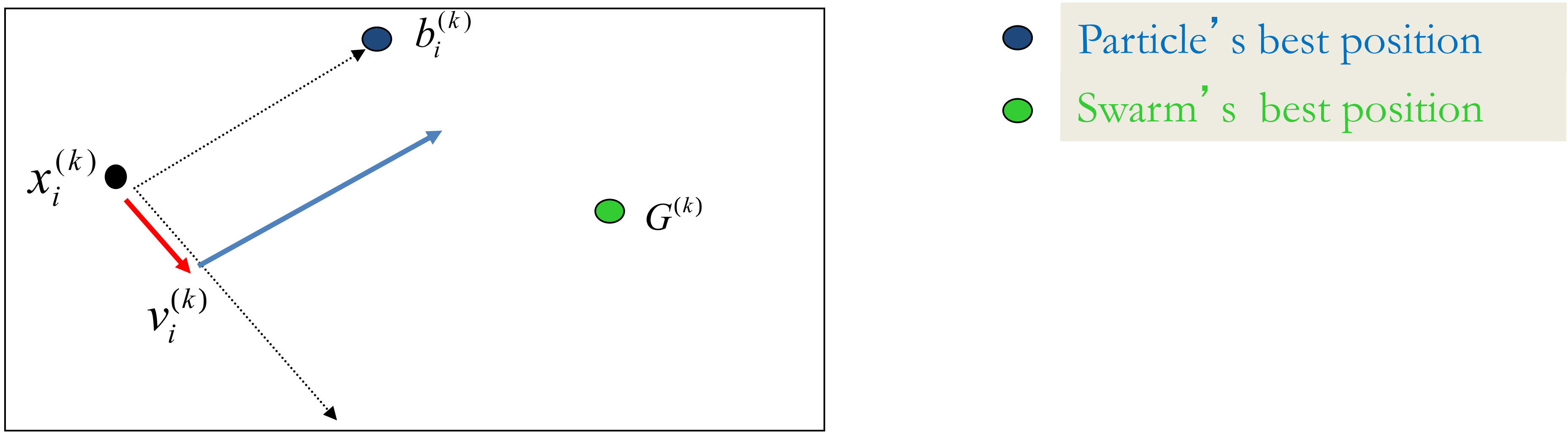
$v_i^{(k)}$ is its velocity (movement)

$b_i^{(k)}$ is the best position visited by the i^{th} particle

$G^{(k)}$ is the overall best position ever visited

w: inertial weight

- Large inertia weight promotes global exploration of the solutions
- Small inertia weight promotes local exploration of the solutions



$$v_i^{(k+1)} = \underbrace{w.v_i^{(k)}} + \underbrace{c_1(b_i^{(k)} - x_i^{(k)})}$$

$x_i^{(k)}$ is the position of the i^th particle at step k

$v_i^{(k)}$ is its velocity (movement)

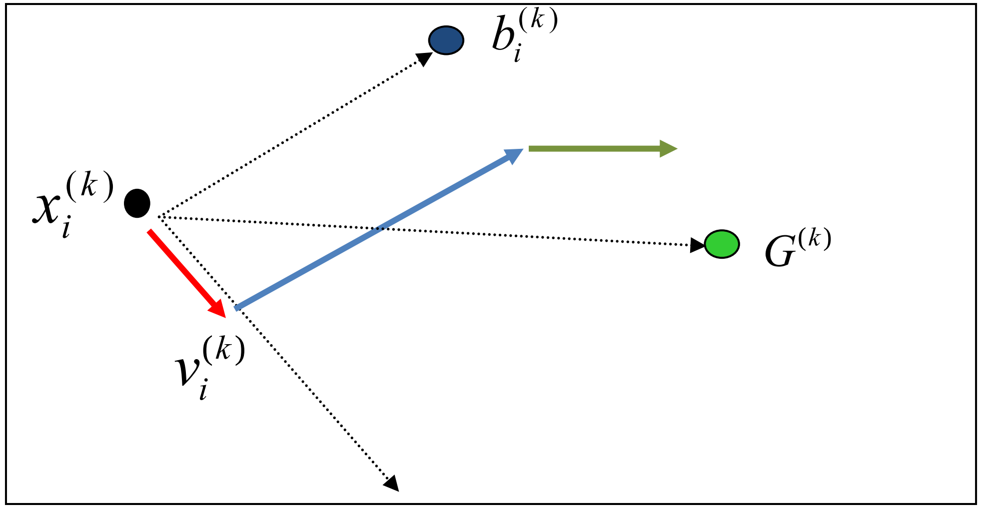
$b_i^{(k)}$ is the best position visited by the i^th particle

$G^{(k)}$ is the overall best position ever visited

w: inertial weight

c_1: Cognitive factor

C_1: represents how much the particle trusts its own past best experience.



Particle's best position

Swarm's best position

$$v_i^{(k+1)} = w.v_i^{(k)} + c_1(b_i^{(k)} - x_i^{(k)}) + c_2(G^{(k)} - x_i^{(k)})$$

$x_i^{(k)}$ is the position of the i[th] particle at step k

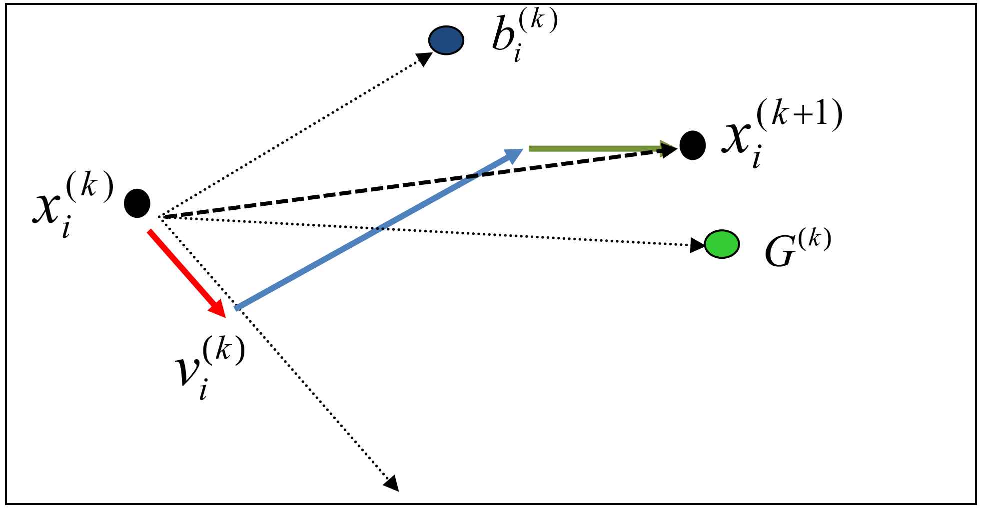
$v_i^{(k)}$ is its velocity (movement)

$b_i^{(k)}$ is the best position visited by the i[th] particle

$G^{(k)}$ is the overall best position ever visited

W: inertial weight
C_1: Cognitive factor
C_2: Social factor

C_1: represents how much the particle trusts its own past best experience.
C2: represents how much the particle trusts the swarm.



Particle's best position

Swarm's best position

$$v_i^{(k+1)} = w.v_i^{(k)} + c_1(b_i^{(k)} - x_i^{(k)}) + c_2(G^{(k)} - x_i^{(k)})$$

$x_i^{(k)}$ is the position of the i^{th} particle at step k

$v_i^{(k)}$ is its velocity (movement)

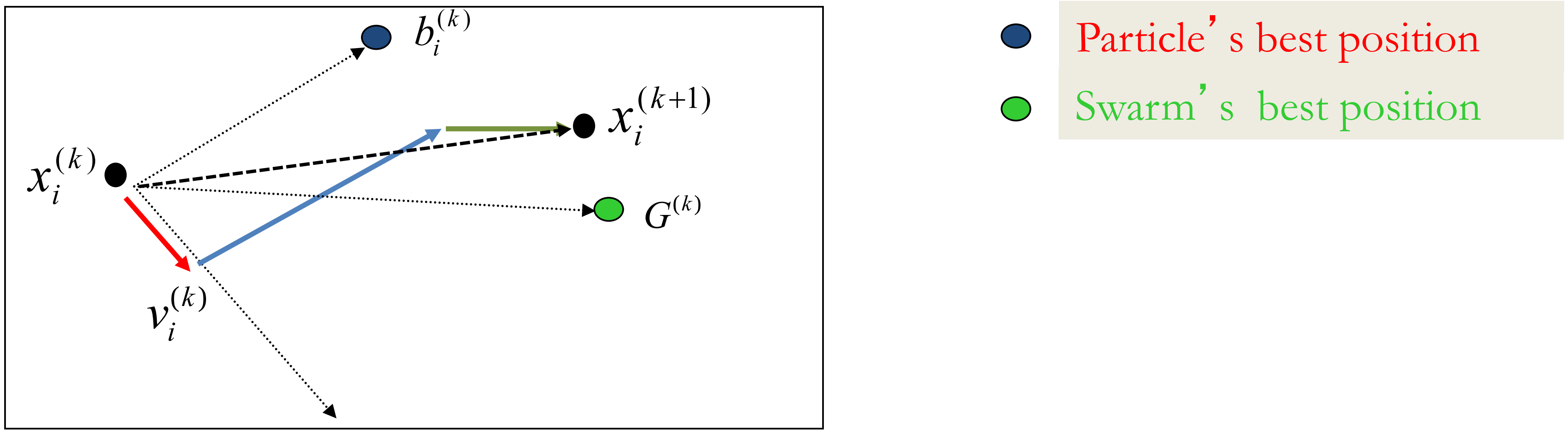
$b_i^{(k)}$ is the best position visited by the i^{th} particle

$G^{(k)}$ is the overall best position ever visited

W: inertial weight
C_1: Cognitive factor
C_2: Social factor

Acceleration coefficient

C_1: represents how much the particle trusts its own past best experience.
C2: represents how much the particle trusts the swarm.



$$v_i^{(k+1)} = w.v_i^{(k)} + c_1 r_1 (b_i^{(k)} - x_i^{(k)}) + c_2 r_2 (G^{(k)} - x_i^{(k)})$$

$x_i^{(k)}$ is the position of the i[th] particle at step k

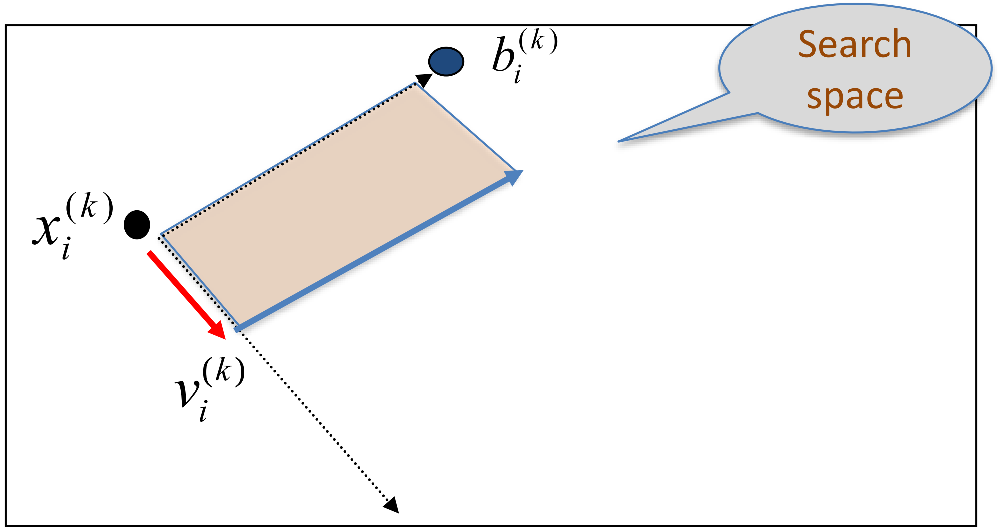
$v_i^{(k)}$ is its velocity (movement)

$b_i^{(k)}$ is the best position visited by the i[th] particle

$G^{(k)}$ is the overall best position ever visited

W: Inertial weight
C_1: Cognitive factor
C_2: Social factor
r_1, r_2: uniform random number in [0,1]

C_1: represents how much the particle trusts its own past best experience.
C2: represents how much the particle trusts the swarm.



Particle's best position

Swarm's best position

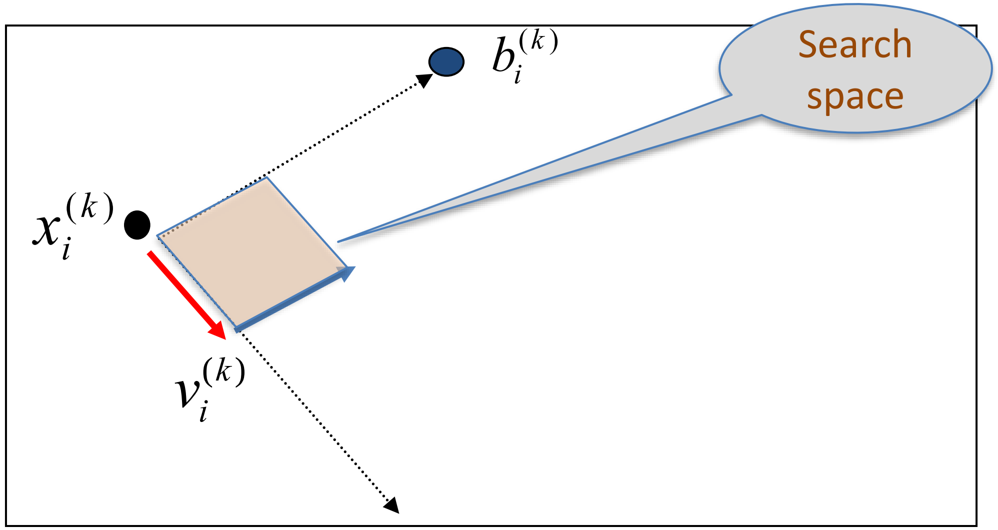
$$v_i^{(k+1)} = w.v_i^{(k)} + c_1 0.7(b_i^{(k)} - x_i^{(k)})$$

W: Inertial weight

C_1: Cognitive factor

C_2: Social factor

r_1, r_2: uniform random number in [0,1]



Particle's best position

Swarm's best position

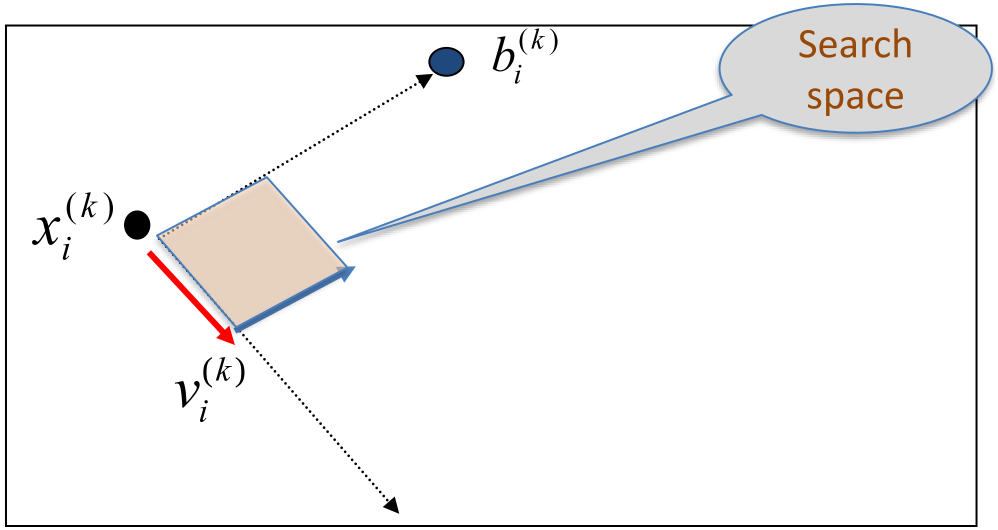
$$v_i^{(k+1)} = w.v_i^{(k)} + c_1 0.2(b_i^{(k)} - x_i^{(k)})$$

W: Inertial weight

C_1: Cognitive factor

C_2: Social factor

r_1, r_2: uniform random number in [0,1]



Particle's best position

Swarm's best position

$$v_i^{(k+1)} = w.v_i^{(k)} + c_1 0.2(b_i^{(k)} - x_i^{(k)})$$

- Large inertia weight promotes global exploration of the solutions
- Small inertia weight promotes local exploration of the solutions
- ***w*** must be selected carefully and/or decreased over iterations.

W: Inertial weight
C_1: Cognitive factor
C_2: Social factor
r_1, r_2: uniform random number in [0,1]

PSO Algorithm

Initialize the particles' positions (x_i), velocity (v_i), previous best positions (b_i), and the number of particles N.

While $t <$ maximum number of iterations (T) **do**

For all Particles **do**

Calculate the fitness $F(x_i)$ function for the current position x_i of the ith particle.

if $F(x_i) > F(b_i)$ **then**

$\boldsymbol{b_i} = x_i$

end if

if $F(x_i) > F(G)$ **then**

$G = x_i$

end if

Update the velocity and positions of all particles according to the Equations.

End for

Stop the algorithm if a sufficiently good fitness function is met.

End while

PSO - Parameters

- Number of particles: usually from 10 to 50
- C_1: importance of personal best
- C_2: importance of neighborhood best
  - Usually $C_1+C_2 = 4$. Given empirically.

Reference

- https://www.youtube.com/watch?v=gkGa6WZpcQg
- https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf
- https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/
- https://www.youtube.com/watch?v=JhgDMAm-imI