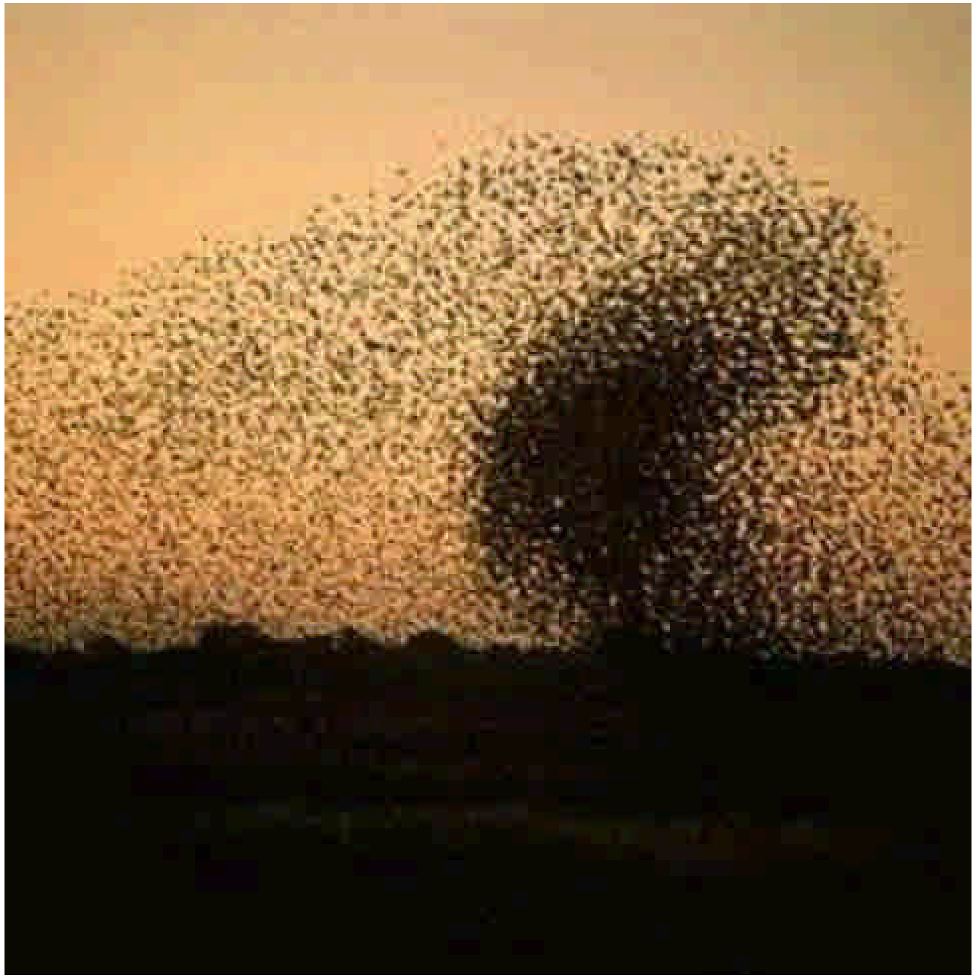
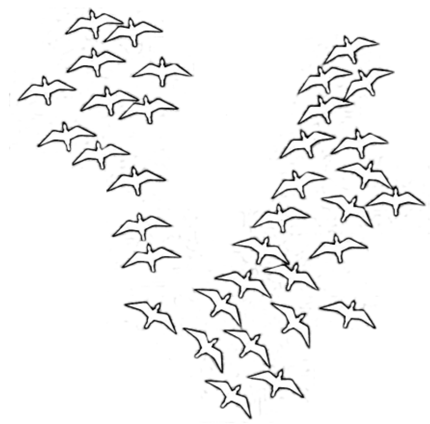
Particle Swarm Optimization





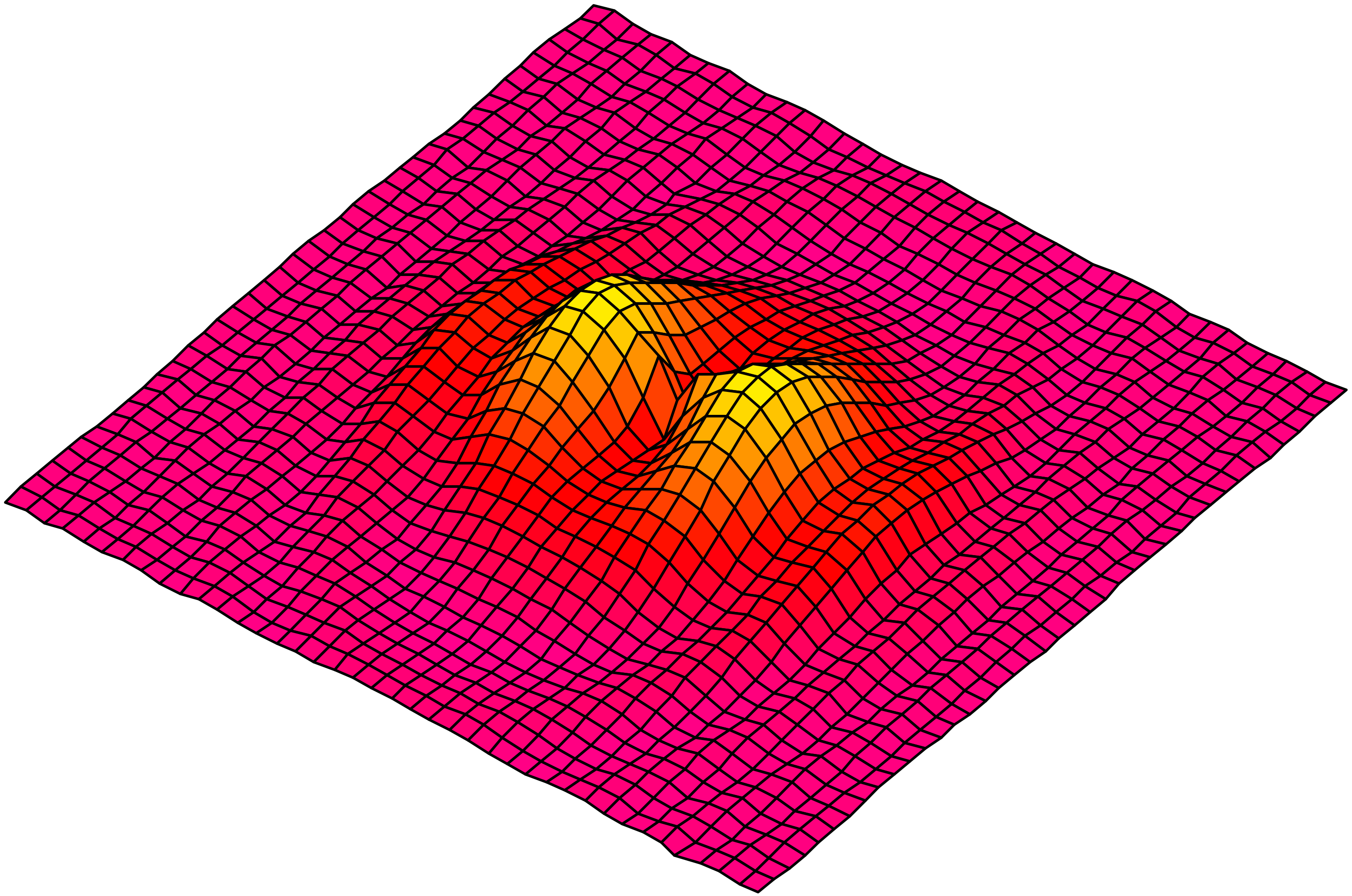
Craig Reynolds(1987) showed that a realistic bird flock could be programmed by implementing three simple rules: match your neighbors' velocity, steer for the perceived center of the flock, and avoid collisions.

Reynolds, C.W. (1987). Flocks, herds, and schools: A distribution behavioral model. *Computer Graphics*, 21, 25-34.

J. Kennedy and R. C. Eberhart, Particle Swarm Optimization. *Proc. of IEEE International Conference on Neural Network*, Piscataway, NJ. Pp.. 1942-1948 (1995).

R. C. Eberhart, and J. Kennedy, A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan. pp. 39-43, 1995.

- Scattering local searches (stochastic search)
- Global Information (population based search algorithm)
- Iterative Evolution (the changes to a particle within the swarm are influenced by the experience, or knowledge, of its neighbors.)



Population-based search algorithm based on the simulation of the social behavior of birds within a flock

Particle Swarm optimization (PSO): In PSO, individuals (potential solutions) are referred as a particles which flown through *hyperdimensional* search space. Changes to the position of particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals. The change to a particle within the swarm are therefore influenced by the experience, or knowledge, of its neighbors

- A *swarm* consists of a set of particles, and each *particle* represents a potential solution.
- Particles fly through the hyperspace, where the position of each particle is changed according to its own experience and that of its neighbors.

$\vec{x}_i(t) =$ position of particle i in hyperspace, at time step t.

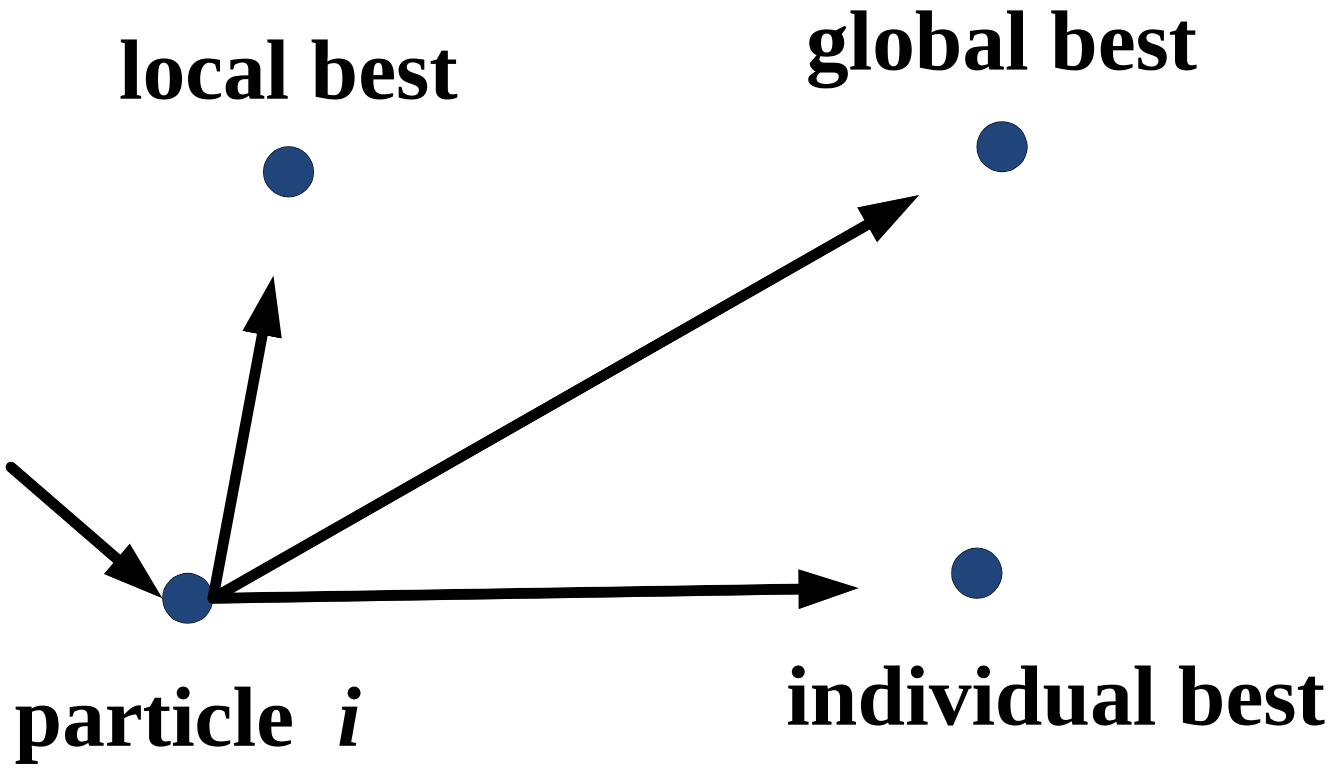
$\vec{v}_i(t) =$ velocity of particle i in hyperspace, at time step t.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$$

$$\vec{x}_i(t) = (x_{i1}(t), x_{i2}(t), \ \ldots \ , x_{in}(t)), \qquad \vec{v}_i(t) = (v_{i1}(t), v_{i2}(t), \ \ldots \ , v_{in}(t)).$$

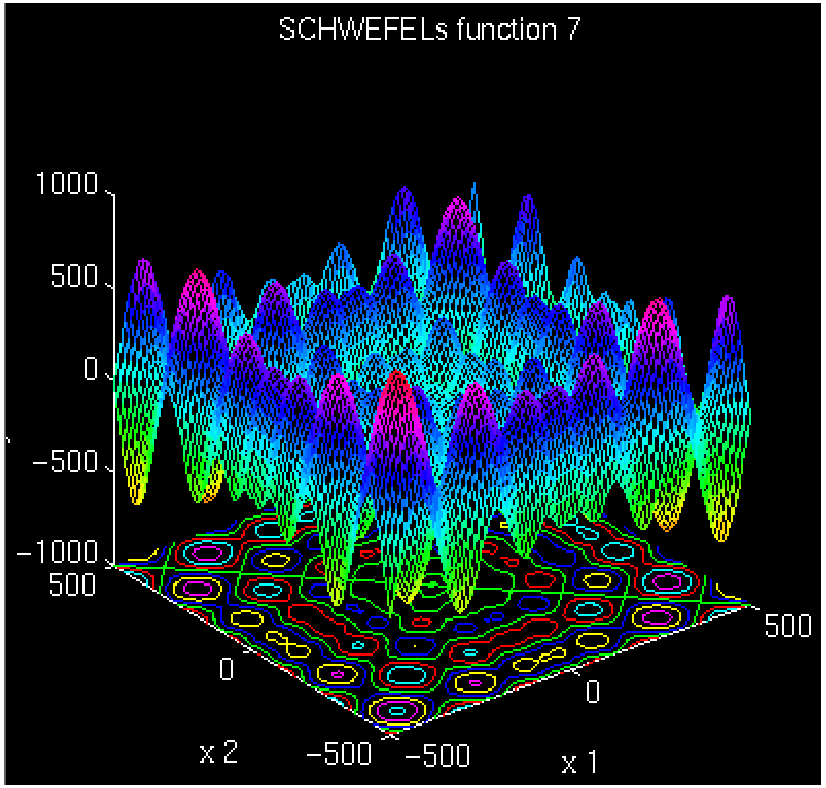
coordinates of particle i at time step t

- The velocity vector derives the optimization process and reflects the social exchanged information.



Different PSO algorithms are different in the extend of the social information exchange.

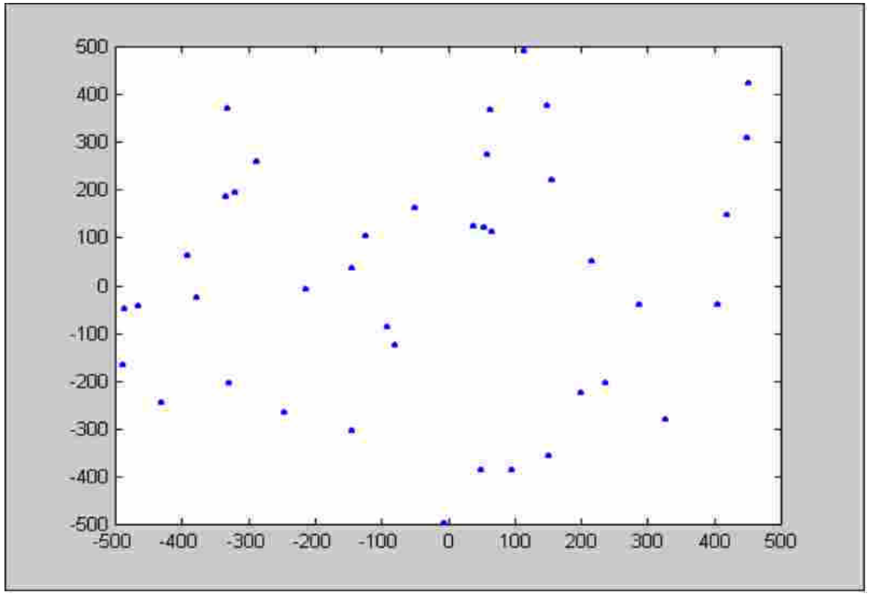
Particle Swarm Optimization (PSO) is a population-based stochastic optimization method proposed by James Kennedy and R. C. Eberhart in 1995. It is motivated by social behavior of organisms such as bird flocking and fish schooling. In the PSO algorithm, the potential solutions called ***particles***, are flown in the problem hyperspace. Change of position of a particle is called ***velocity***. The particle changes their position with time. During flight, particle's velocity is *stochastically* accelerated toward its previous best position and toward a neighborhood best solution. POS has been successfully applied to solve various optimization problems, artificial neural network training, fuzzy system control, and others.



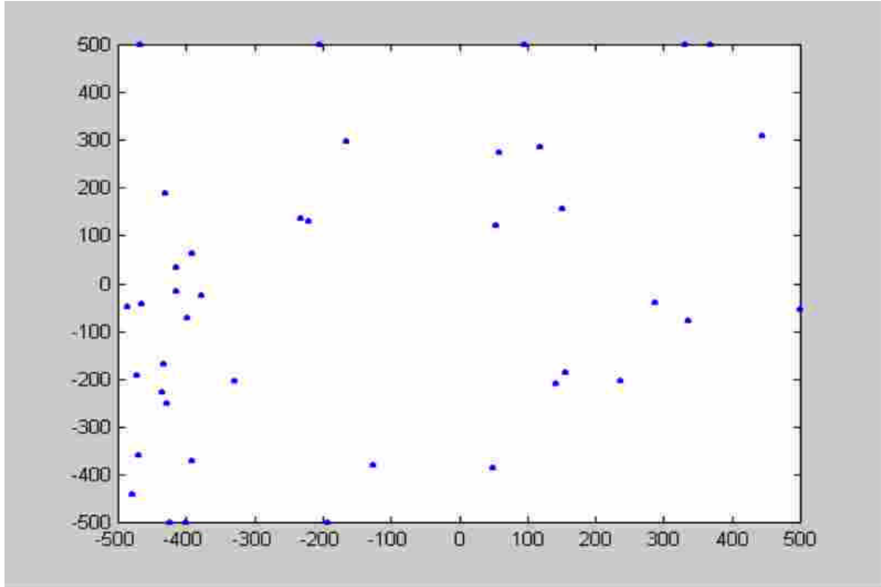
$$f(x) = \sum_{i=1}^{n} (-x_i) \cdot \sin(\sqrt{|x_i|}), \text{ where } -500 \le x_i \le 500$$

global minimum

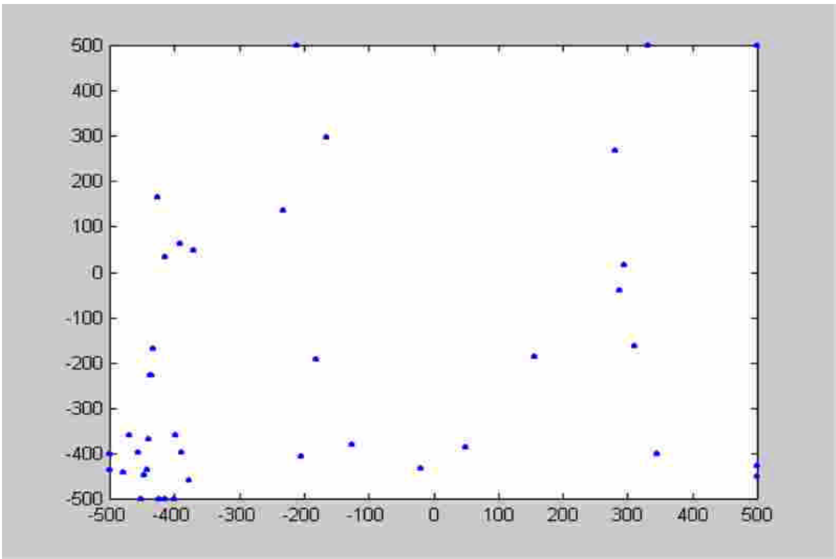
$$f(x) = n \cdot 418.9829; \quad x_i = -420.9687, \; i = 1, \ldots, n$$



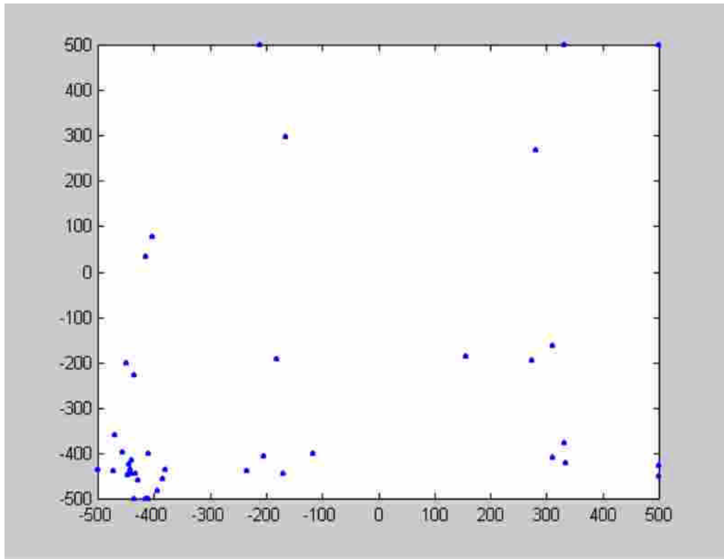
Initial



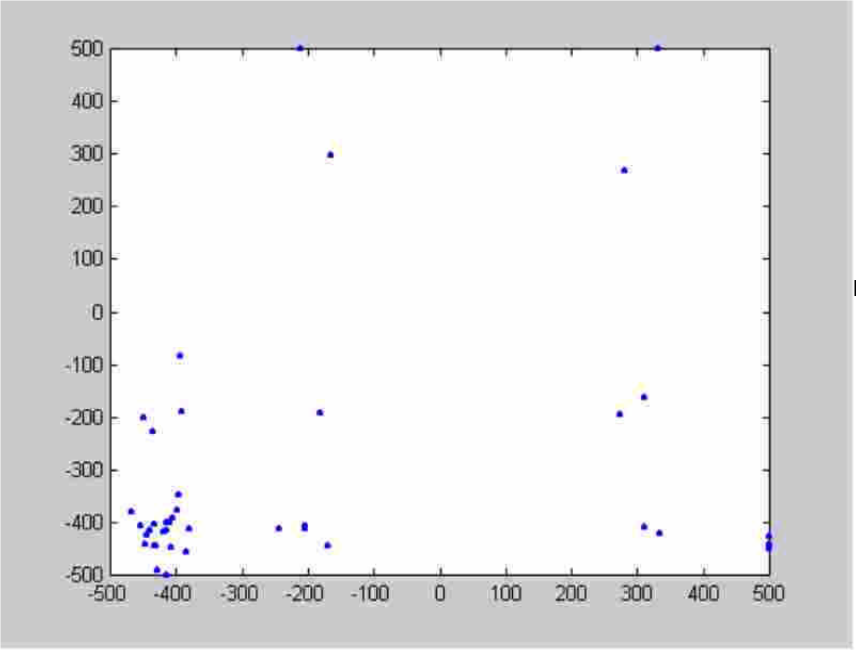
5 generations



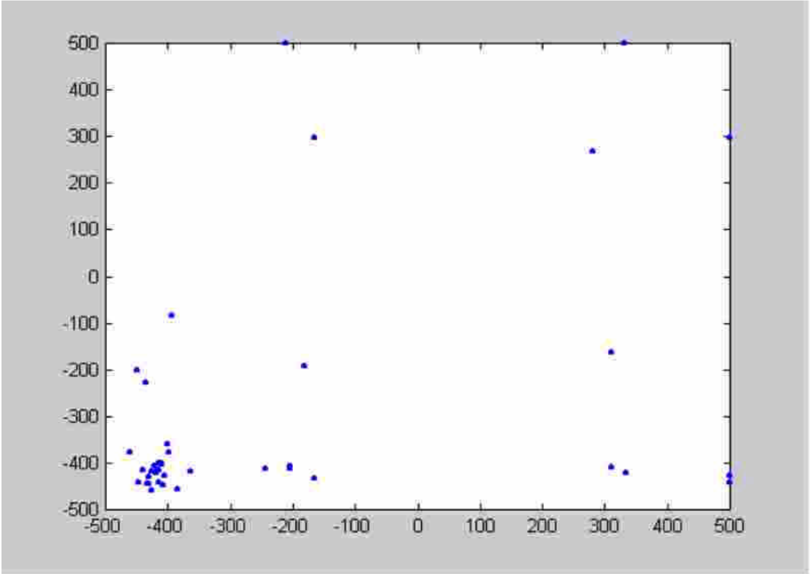
10 generations



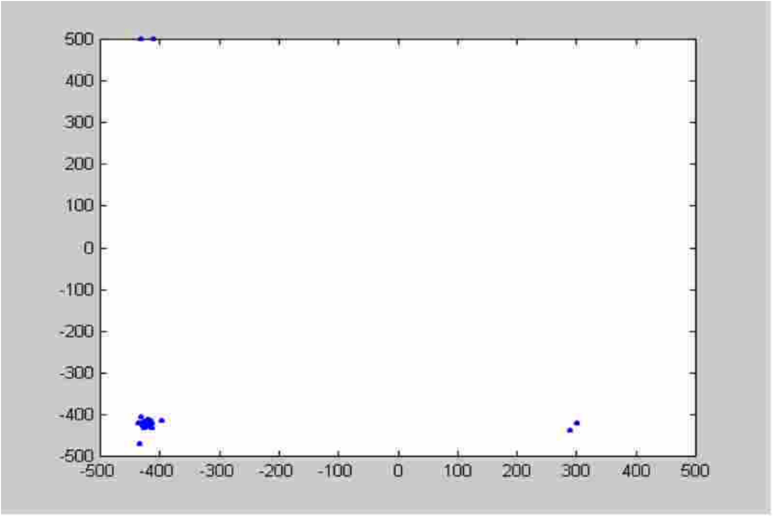
15 generations



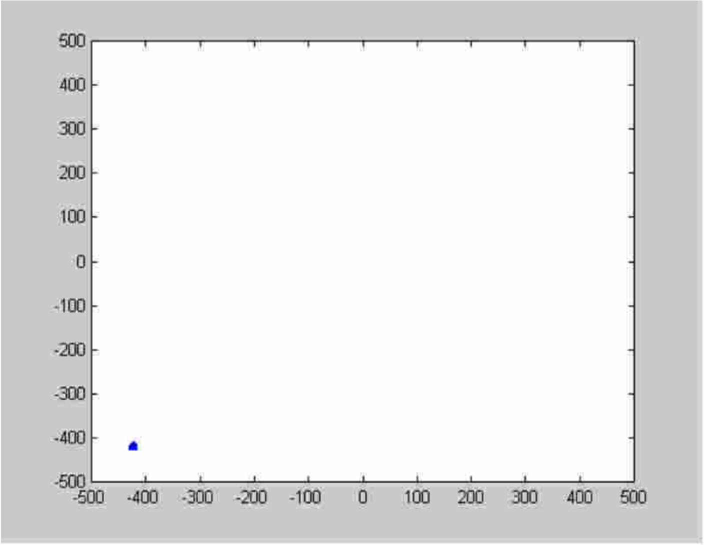
20 generations



25 generations

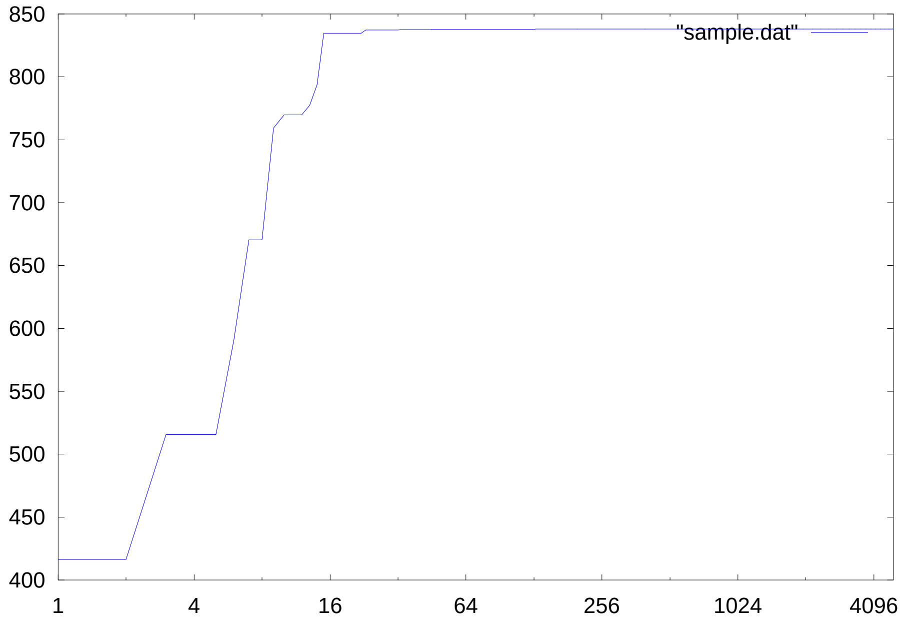


100 generations



500 generations

iterations	$f(x_1, x_2)$
0	416.245599
5	515.748796
10	759.404006
15	793.732019
20	834.813763
100	837.911535
5000	837.965771
optimal	837.9658



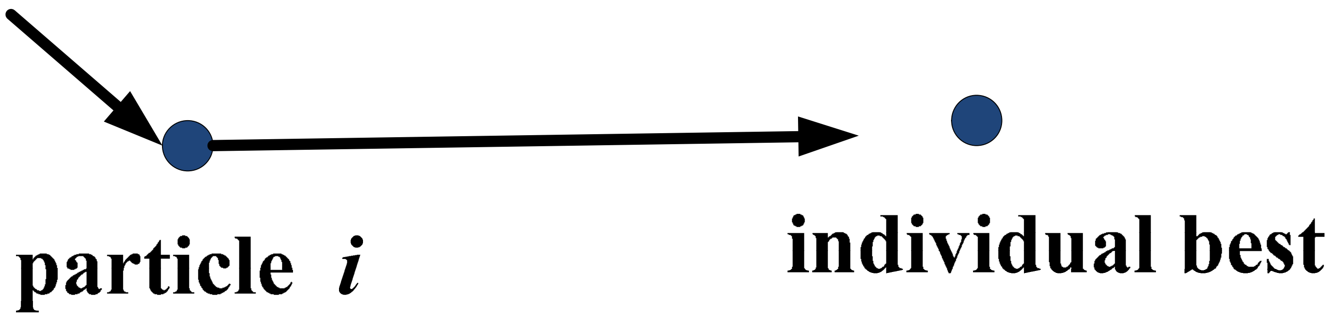
- Kennedy and Eberhart 1995 aim to discover the patterns that govern the ability of birds to fly synchronously, and to suddenly change the direction with a regrouping in an optimal formation.

J. Kennedy and R.C. Eberhart, Particle Swarm Optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Vol.4, pp. 1942-1948, 1995

. Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan. pp. 39-43, 1995

- a population-based search algorithm
- based on the simulation of the social behavior of birds within a flock.
- the changes to a particle within the swarm are influenced by the experience, or knowledge, of its neighbors.

Individual Best Algorithm:



Each particle compares its current position to its own best position, pbest, only. No information from other particles is used.

Different PSO algorithms are different in the extend of the social information exchange.

Individual Best Algorithm:

Each particle compares its current position to its own best position, pbest, only. No information from other particles is used.

1. (**Initialization**) At $t = 0$, the swam P(0) = { $P_1, P_2, \ . \ . \ . \ , P_k$ }. For, $i = 1, \ldots k$, the position $\vec{x}_i(0)$ of particle $P_i \in$ P(0) is random within the hyperspace. Initial velocity $\vec{v}_i(0)$ of particle P_i is given for each i.
   (We assume that the swarm has k particles.)

2. (**Evaluation of Particles**) Evaluate the performance of each particle, using its current position $\vec{x}_i(t)$. $F(\vec{x}_i(t))$ is the fitness of particle i at time step t.

3. (**Comparison**) Compare the performance of each particles to its best performance thus far:

   If $F(\vec{x}_i(t)) < pbest_i$, then

   $(a) \quad pbest_i = F(\vec{x}_i(t))$ and $(b) \quad \vec{x}_{pbest_i} = \vec{x}_i(t)$.

4. (**Change the Velocity Vector**) Change the velocity vector for each particle as follows:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho(\vec{x}_{pbest_i} - \vec{x}_i(t))$$

where ρ is a positive random number.

5. (**Move to a New Position**) Move each particle to a new position.

$(a)\quad \vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$

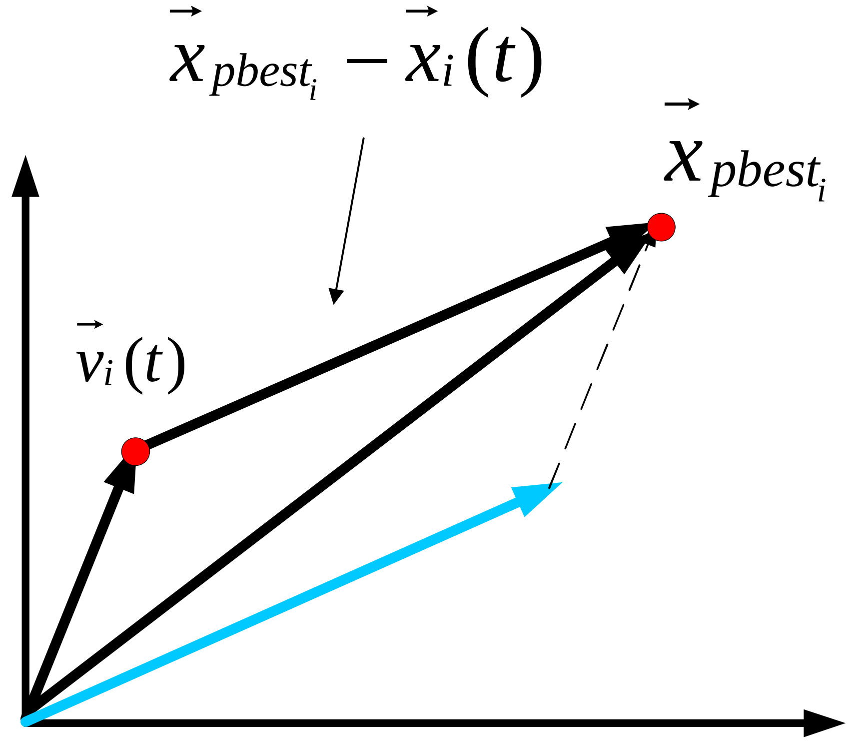
$(b)\quad t \leftarrow t+1$

New Position by Random Steps

6. Go to step 2, and repeat until convergence.

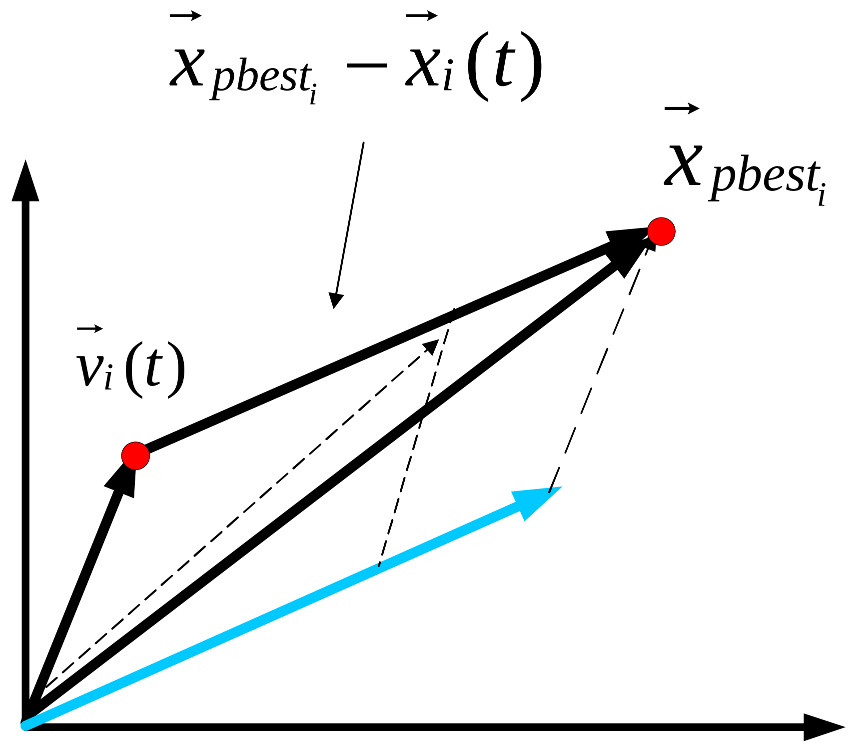
$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho(\vec{x}_{pbest_i} - \vec{x}_i(t))$$

where ρ is a positive random number

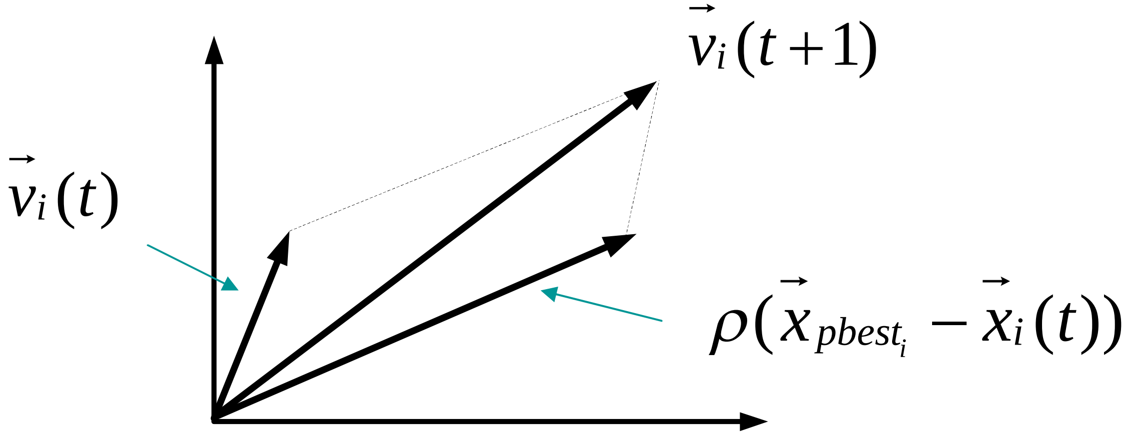


$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho(\vec{x}_{pbest_i} - \vec{x}_i(t))$$

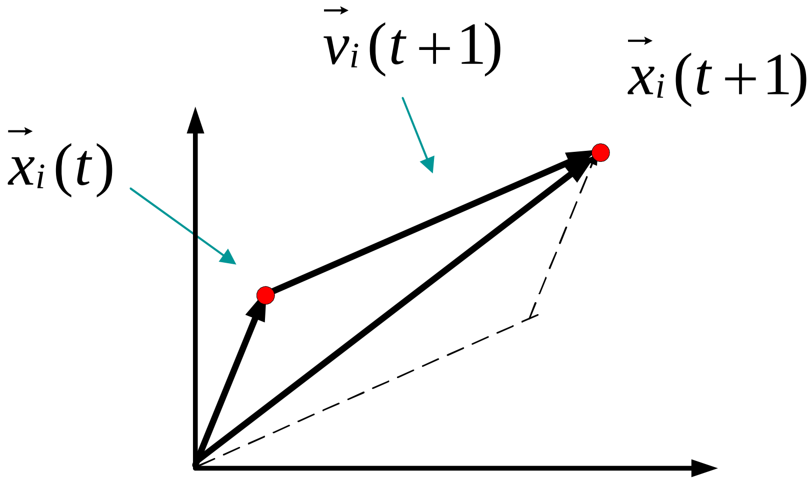
where ρ is a positive random number



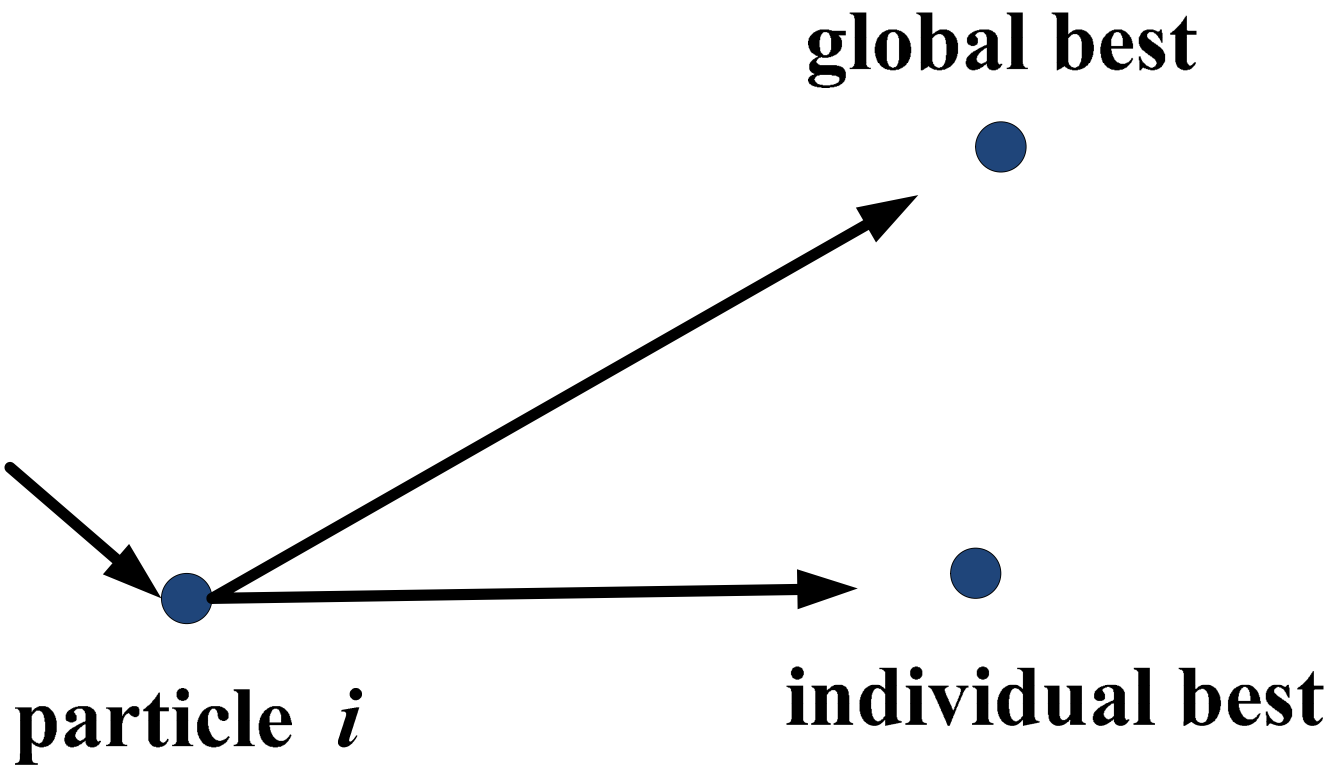
$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho(\vec{x}_{pbest_i} - \vec{x}_i(t))$$



$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$$

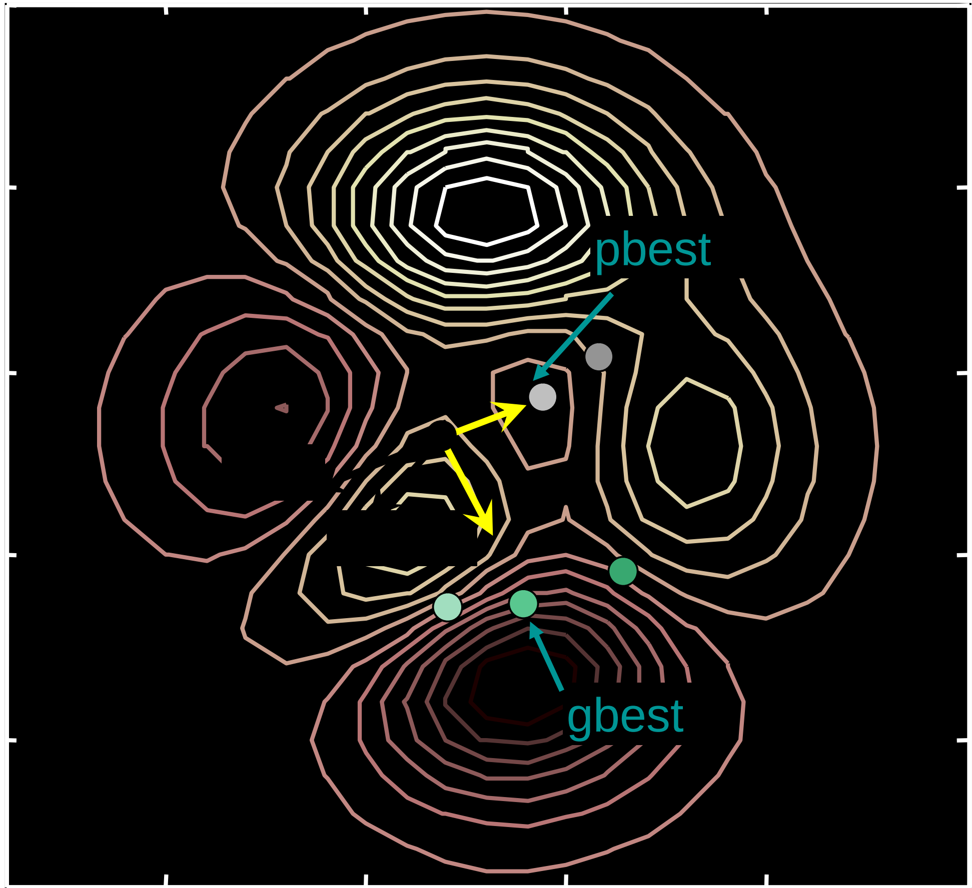


Global Best Algorithm:



Each particle compares its current position to the entire swarm best position, *gbest*.

Global Best Algorithm:



Global Best Algorithm:

Each particle compares its current position to the entire swarm best position, *gbest*.

1. (**Initialization**) At $t = 0$, the swam P(0) = { $P_1, P_2, \ldots, P_k$ }. For, $i = 1, \ldots k$, the position of particle $P_i \in$ P(0), $\vec{x}_i(0)$ is random within the hyperspace and initial velocity $\vec{v}_i(0)$ of particle P_i is given for each i.
   (We assume that the swarm has k particles.)

2. (**Evaluation of Particles**) Evaluate the performance of each particle, using its current position $\vec{x}_i(t)$. $F(\vec{x}_i(t))$ is the fitness of particle i at time step t.

3. (**Comparison**) Compare the performance of each particles to its best performance thus far:

   If $F(\vec{x}_i(t)) < pbest_i$, then

   (a) $pbest_i = F(\vec{x}_i(t))$ and (b) $\vec{x}_{pbest_i} = \vec{x}_i(t)$.

Compare the performance of each particles to the global best particle thus far:

If $F(\vec{x}_i(t)) < gbest$, then

$(a) \quad gbest = F(\vec{x}_i(t))$

$(b) \quad \vec{x}_{gbest} = \vec{x}_i(t)$

Move between individual & global Best

cognitive component

social component

4. (**Change the Velocity Vector**) Change the velocity vector for each particle as follows:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_i(t))$$

where ρ_1 an ρ_2 are positive random numbers.

5. (**Move to a New Position**) Move each particle to a new position.

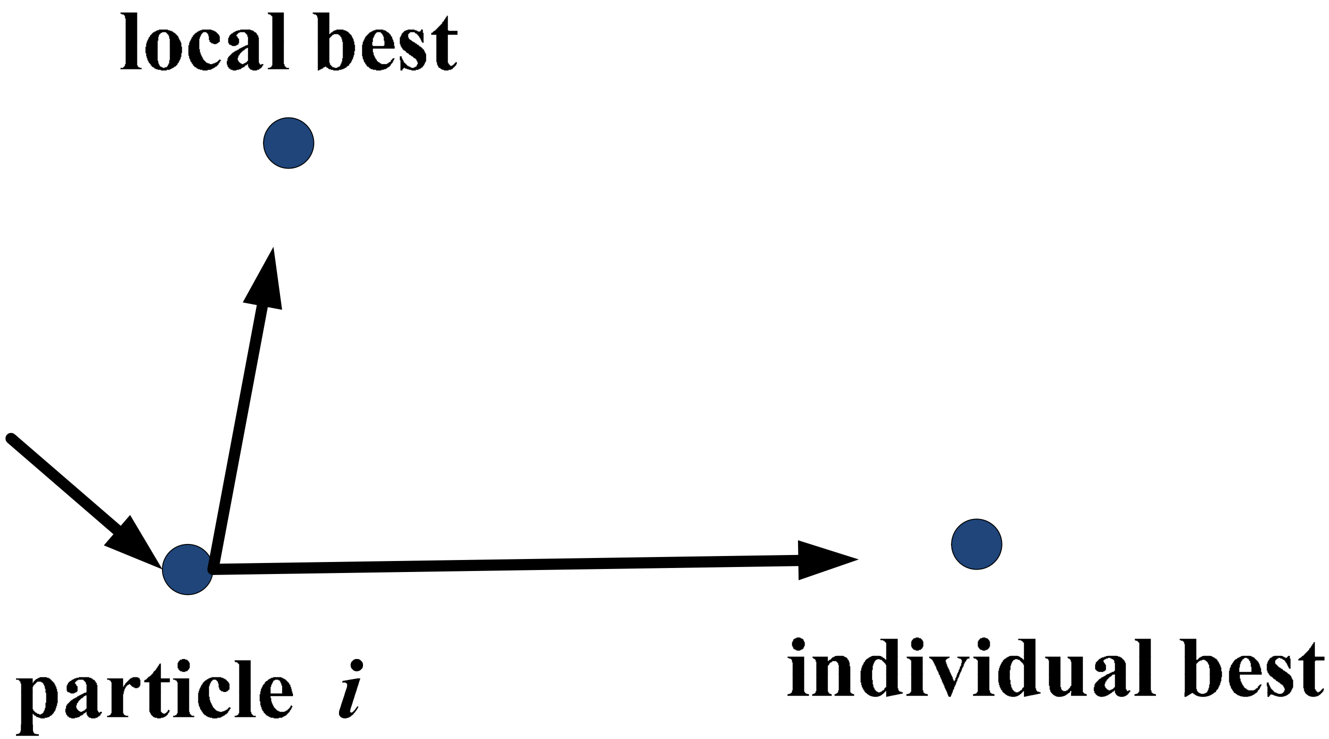
$(a) \quad \vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$

$(b) \quad t \leftarrow t+1$

$$\vec{v}_i(t+1) = \phi\vec{v}_i(t) + \rho_1(\vec{x}_{pbest} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_i(t))$$

6. Go to step 2, and repeat until convergence.

Local Best Algorithm:



Particles are influenced by the *lbest*, the best position within their neighborhood, as well as their own past experience.

Local Best Algorithm: generalization of **Global Best Algorithm**

In the local best algorithm, it reflects the circle neighborhood structure. Particles are influenced by the *lbest*, the best position within their neighborhood, as well as their own past experience. In Global Best Algorithm, only steps 3 and 4 are changed by replacing *gbest* with *lbest*.

1. (**Initialization**) At $t = 0$, the swam P(0) = { $P_1, P_2, \ldots, P_k$ }. For, $i = 1, \ldots k$, the position of particle $P_i \in$ P(0), $\vec{x}_i(0)$ is random within the hyperspace and initial velocity $\vec{v}_i(0)$ of particle P_i is given for each i.
   (We assume that the swarm has k particles.)

2. (**Evaluation of Particles**) Evaluate the performance of each particle, using its current position $\vec{x}_i(t)$. $F(\vec{x}_i(t))$ is the fitness of particle i at time step t.

3. (**Comparison**) Compare the performance of each particles to its best performance thus far:

   If $F(\vec{x}_i(t)) < pbest_i$, then

   (a) $pbest_i = F(\vec{x}_i(t))$ and (b) $\vec{x}_{pbest_i} = \vec{x}_i(t)$.

Compare the performance of each particles to the local best particle thus far:

If $F(\vec{x}_i(t)) < lbest$, then

$(a)\quad lbest = F(\vec{x}_i(t))$

$(b)\quad \vec{x}_{lbest} = \vec{x}_i(t)$

cognitive component

social component

4. (**Change the Velocity Vector**) Change the velocity vector for each particle as follows:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{lbest} - \vec{x}_i(t))$$

where ρ_1 an ρ_2 are positive random numbers.

5. (**Move to a New Positon**) Move each particle to a new position.

$(a)\quad \vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1)$

$(b)\quad t \leftarrow t+1$

6. Go to step 2, and repeat until convergence.

Remark:

(1) The random numbers ρ_1 and ρ_2 are defined as

$$\rho_1 = r_1 c_1 \text{ and } \rho_2 = r_2 c_2,$$

where $r_1, r_2 \in U(0, 1)$, and c_1 and c_2 are acceleration constants. Kennedy has studied the effect of the random variables ρ_1 and ρ_2 on the particle trajectories, and asserted that $c_1 + c_2 \leq 4$ [Kennedy 1998]. If $c_1 + c_2 > 4$, velocities and positions explode toward infinity.

J. Kennedy, The Behavior of Particles, in V.W. Porto, N. Saravana, D. Waagen (ed.), *Proceedings of the 7th Conference on Evolutionary Programming, 1998*, pp. 581-589.

(2) Fitness Calculation: Fitness function is to measure the performance of each particle. Hence, a function is used to measure the closeness of the corresponding solution to the optimum.

(3) Convergence: The following criterion may be used for termination of the PSO algorithms.

(a) PSO algorithm is executed for a fixed number of iterations.

(b) PSO algorithm can be terminated if the velocity changes are close to zero for all particles, in which case there will be no further changes in particle positions.

(4) Parameters: Standard PSO algorithm is influenced by the six system parameters.

(a) dimension of the problem.

(b) number of particles in each iteration), *k*.

(c) upper limit of ρ.

(d) upper limit on the maximum velocity.

(e) the neighborhood size.

(f) Inertia weights.

Maximum velocity, A upper limit is placed on the velocity in all dimensions. It prevents particles from moving too rapidly from one region in search space to another.

$$\text{If } v_{ij}(t) > V_{max} \text{ then } v_{ij}(t) = V_{max}, \, or$$

$$\text{If } v_{ij}(t) < -V_{max} \text{ then } v_{ij}(t) = -V_{max}.$$

$$\vec{v}_i(t) = (v_{i1}(t), v_{i2}(t), \ \dots, v_{in}(t))$$

Neighborhood size, The *gbest* version is the *lbest* with the entire swarm as the neighborhood. The *gbest* is more susceptible to local optimum, since all particles are pulled toward that solution. The smaller the neighborhood size, and the more neighborhoods can be used, the less susceptible PSO is to local optimum. A larger part of search space is traversed, and no one particle has an influence on all particles. The larger the neighborhood size is, the slower the algorithm converges.

Inertia weight,

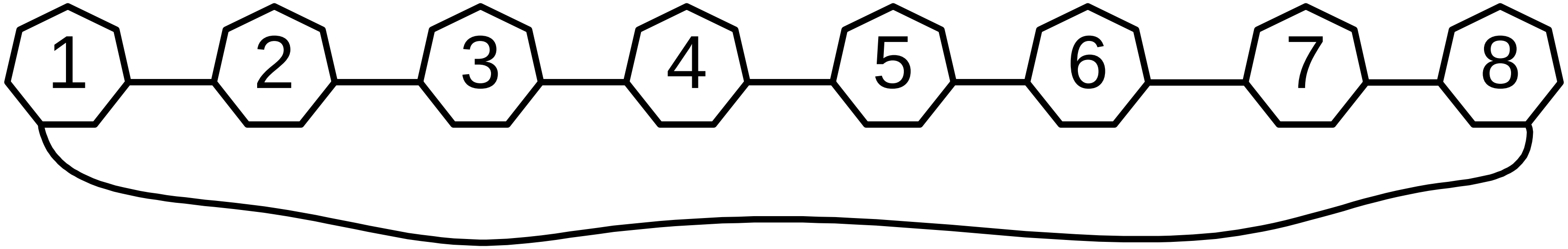
$$\vec{v}_i(t+1) = \phi\vec{v}_i(t) + \rho_1(\vec{x}_{pbest} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_i(t))$$

where ϕ is the inertia weight. It controls the influence of previous velocities on the new velocity. Larger inertia weight cause larger exploration of the search space, while smaller inertia weights focus the search on smaller region. Typically, PSO is started with a large inertia weight, which is decreased over time.

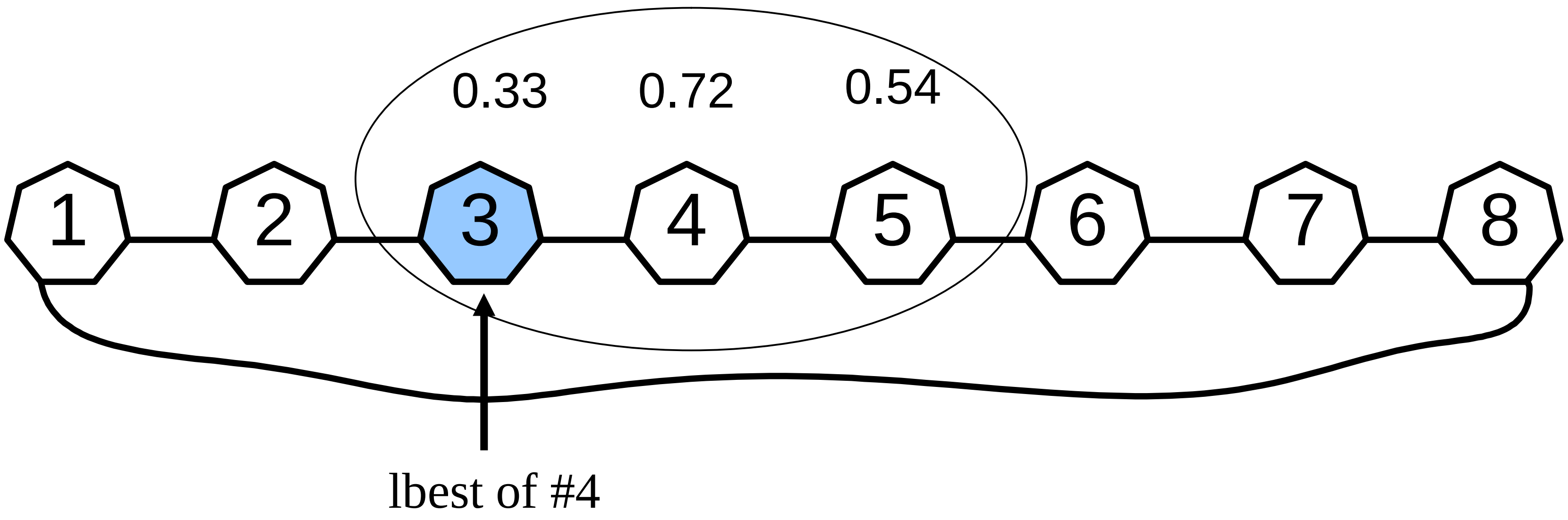
The “lbest” neighbohood with size = 2.

Neighborhood of #**4** is { #**3**, #**5**}.

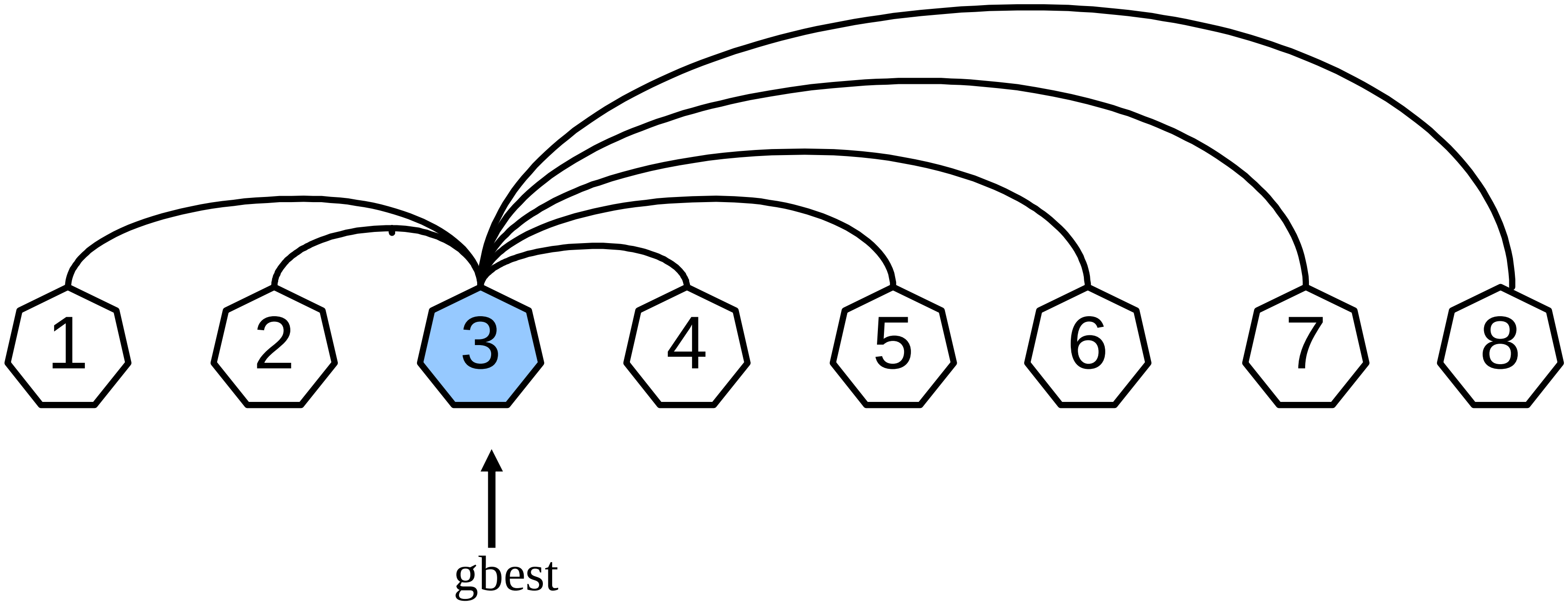
Neighborhood of #**8** is { #**7**, #**1**}.



Particle #3 has found the best position so far in #4's neighbohood. Thus, #4's velocity will be adjusted toward #3's previous best position and #4's own previous best position.



The “gbest” neighborhood. Particle #3 has found the best position so far in the entire population, all other’s velocities will be attracted toward its previous best position.



A Discrete Version of Global Best Algorithm: (Binary PSO)

1. (**Initialization**) At t = 0, the swam P(0) = { P_1, P_2, . . . , P_k }. For, i = 1, … k, the position of particle $P_i \in$ P(0), $\vec{x}_i(0)$ is random within the hyperspace and initial velocity $\vec{v}_i(0)$ of particle P_i is given for each i.
(We assume that the swarm has k particles.)

2. (**Evaluation of Particles**) Evaluate the performance of each particle, using its current position $\vec{x}_i(t)$. $F(\vec{x}_i(t))$ is the fitness of particle i at time step t.

3. (**Comparison**) Compare the performance of each particles to its best performance thus far:

   If $F(\vec{x}_i(t)) < pbest_i$, then

   (a) $pbest_i = F(\vec{x}_i(t))$ and (b) $\vec{x}_{pbest_i} = \vec{x}_i(t)$.

   Compare the performance of each particles to the global best particle thus far:

   If $F(\vec{x}_i(t)) < gbest$, then

   (a) $gbest = F(\vec{x}_i(t))$, (b) $\vec{x}_{gbest} = \vec{x}_i(t)$.

cognitive component — social component

4. (**Change the Velocity Vector**) Change the velocity vector for each particle as follows:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + \rho_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_i(t))$$

where ρ_1 an ρ_2 are positive random numbers and

$$-v_{max} < v_{ij}(t+1) < v_{max}, \quad j = 1, \dots, n.$$

5. (**Move to a New Position**) Move each particle to a new position.

If $\rho_{ij} < s(v_{ij}(t+1))$, then $x_{ij}(t+1) = 1$, else $x_{ij}(t+1) = 0$,

where $s(v_{ij}(t+1)) = \dfrac{1}{1+\exp(-v_{ij}(t+1))}$. $\rho_{ij} \in U(0, 1)$.

Set $t \leftarrow t+1$.

6. Go to step 2, and repeat until convergence.