# Particle Swarm Optimization

# Particle Swarm Optimization

- **The particle swarm algorithm imitates human social behavior.**

—James Kennedy and Russell Eberhart [Kennedy and Eberhart, 2001]

- Collective intelligence in many natural systems.
  - Flocks of animals as they avoid predators, seek food, seek to travel more quickly, and other behaviors.
- Ants exhibit an extraordinary level of collective intelligence (Ant Colony algorithm)

# Particle Swarm Optimization

- Animal groups can often avoid predators more effectively in a group than alone.
- For example, it might be easy for a lion to recognize a single zebra because of its contrast with the surrounding landscape, but a group of zebras blend together and are more difficult to recognize as individuals.

# Particle Swarm Optimization

- A group of animals might also appear to be larger, or sound louder, or be more threatening in other ways, than a solitary animal. Finally, it may be difficult for a predator to focus on a single animal when it is part of a large group. These phenomena are called the **predator confusion effect** [Milinski and Heller, 1978]

# Particle Swarm Optimization

- Another way that groups protect themselves from predators is described by the **many-eyes hypothesis** [Lima, 1995].
- When a large group forages for food or drinks from a stream, random effects dictate that there will always be a few animals who are watching for predators. This collaboration not only provides more protection from predators, but also allows each individual more time for feeding and drinking.

# Particle Swarm Optimization

- Animals also have more success in finding food when they are in groups than when they are alone.

# Particle Swarm Optimization

- Animals can also move more quickly when in groups than when alone.
- The trailing riders might expend as much as 40% less energy than the lead rider because of wind resistance




# Particle Swarm Optimization

- Particle swarm optimization (PSO) is based on the observation that groups of individuals work together to improve not only their collective performance on some task, but also each individual performance.
- The principles of PSO are clearly seen not only in animal behavior but also in human behavior.

# Particle Swarm Optimization

- As we try to improve our performance at some task, we adjust our approach based on some basic ideas.
- **Inertia.**
  - We tend to **stick to the old ways** that have proven to be successful in the past.
  - "I've always done it this way, and so that is how I am going to continue doing things.

# Particle Swarm Optimization

- **Influence by society.**
  - We hear about others who have been successful and we try to emulate their approaches.
  - We may read about the success of others in books, or on the internet, or in the newspaper.
  - "If it worked for them, then maybe it will work for me too.“

# Particle Swarm Optimization

- **Influence by neighbors**.
  - We learn the most from those who are personally close to us.
  - We are influenced more by our friends than by society. In our conversations with others, we share stories of success and failure, and we modify our behavior because of those conversations.
  - Investment advice from our millionaire neighbor or cousin will have a stronger influence on us than the more distant stories of billionaires that we read on the internet.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- Suppose that we have a **minimization problem** that is defined over a continuous domain of **d dimensions**.
- We also have a **population of N candidate** solutions, denoted as {xi}, i ϵ [1, N].
- Furthermore, suppose that **each individual xi is moving with some velocity Vi through the search space**.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- As a PSO individual moves through the search space, it has some **inertia** and so it tends to maintain its velocity.
- However, its velocity can change due to a couple of different factors.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- However, its velocity can change due to a couple of different factors.
- First, it **remembers its best position in the past**, and it would like to change its velocity to return to that position. This is similar to the human tendency to remember the good old days, and to try to recapture the experiences of the past.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- However, its velocity can change due to a couple of different factors.
- Second, an **individual knows the best position of its neighbors at the current generation**.
  - This requires the definition of a neighborhood size, and it requires that **all of the neighbors communicate with each other** about their performance on the optimization problem.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

Initialize a random population of individuals $\{x_i\}$, $i \in [1, N]$
Initialize each individual's $n$-element velocity vector $v_i$, $i \in [1, N]$
Initialize the best-so-far position of each individual: $b_i \leftarrow x_i$, $i \in [1, N]$
Define the neighborhood size $\sigma < N$
Define the maximum influence values $\phi_{1,\max}$ and $\phi_{2,\max}$
Define the maximum velocity $v_{\max}$
While not(termination criterion)
    For each individual $x_i$, $i \in [1, N]$
        $H_i \leftarrow \{\sigma$ nearest neighbors of $x_i\}$
        $h_i \leftarrow \arg\min_x \{f(x) : x \in H_i\}$
        Generate a random vector $\phi_1$ with $\phi_1(k) \sim U[0, \phi_{1,\max}]$ for $k \in [1, n]$
        Generate a random vector $\phi_2$ with $\phi_2(k) \sim U[0, \phi_{2,\max}]$ for $k \in [1, n]$
        $v_i \leftarrow v_i + \phi_1 \circ (b_i - x_i) + \phi_2 \circ (h_i - x_i)$
        If $|v_i| > v_{\max}$ then
            $v_i \leftarrow v_i v_{\max} / |v_i|$
        End if
        $x_i \leftarrow x_i + v_i$
        $b_i \leftarrow \arg\min\{f(x_i), f(b_i)\}$
    Next individual
Next generation

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- **Initialize velocities**- initialized randomly, or they could be initialized to zero.
- Smaller **neighborhoods** (as small as two) provide better global behavior and avoid local minima, while larger neighborhoods provide faster convergence.
- We have to choose the maximum **learning rates** ϕ1max and ϕ2max
- **Φ1**- **cognition learning rate**
- **Φ2**- **Social learning rate**
  - Random numbers distributed in [0, ϕ1max] and [0, ϕ2max] respectively.

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

- We have to choose the **maximum velocity Vmax**. Empirical evidence indicates that each element of **Vmax should be limited to the corresponding dynamic range of the search space** [Eberhart and Shi, 2000].
- If Vmax were greater than the dynamic range of the search space, then a **particle could easily leave the search space** in a single generation.
- Other results suggest setting Vmax to between 10% and 20% of the search space range [Eberhart and Shi, 2001].

# A BASIC PARTICLE SWARM OPTIMIZATION ALGORITHM

$$v_i \leftarrow v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)$$

**Φ1**- **cognition learning rate**
**Φ2**- **Social learning rate**
**bi – Best performance in past experience.**
**hi – Best neighbour**

$$\begin{aligned} x_i &\leftarrow \min(x_i, x_{\max}) \\ x_i &\leftarrow \max(x_i, x_{\min}) \end{aligned}$$

[Xmin, Xmax] defines the limits of the search domain.

# Particle Swarm Topologies

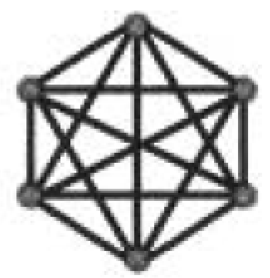
**The arrangement of the neighbors that influence a particle is call the *topology of the* swarm.**

If the neighborhood of each particle changes each generation, it is called a **dynamic topology**.
If the neighborhood is local (that is, it does not include the entire swarm), it is also called an **lbest topology**.

We could define neighborhoods at the beginning of the algorithm so that the **neighborhoods are static and do not change from one generation to the next.** Or, if the optimization process stagnates, we could **at that time randomly redefine the neighborhoods.**

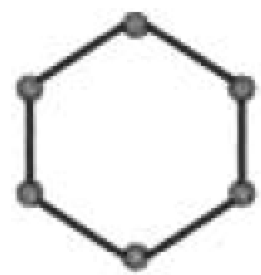
# Particle Swarm Topologies

- In the extreme case **we can have a single neighborhood that encompasses the entire swarm**, which means that Hi is equal to the entire swarm for all i, and hi is independent of i and is equal to the best particle among the entire population.
- This is called the **all topology** or the **gbest topology**.



# Particle Swarm Topologies

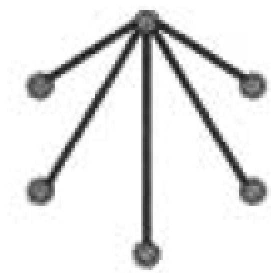
- Another common topology is the **ring topology**, in which each particle is connected to two other particles.



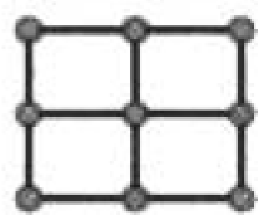
- The **cluster topology** is one in which each particle is fully connected within its own cluster, while a few particles in each cluster are also connected to an additional particle in another cluster.

# Particle Swarm Topologies

- The **wheel topology** is one in which a focal particle is connected to all other particles, while all of the other particles are connected only to the focal particle.



- The **square topology**, also called the **von Neumann topology**, is one in which each particle is connected to four neighbors.



# VELOCITY LIMITING

- It has been found in many applications of PSO that if **Vmax is not used**, PSO **particles jump wildly** around the search space [Eberhart and Kennedy, 1995].

$$\begin{aligned} v_i(t+1) &= v_i(t) + \phi_1(b_i - x_i) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned}$$

$$\text{If } |v_i| > v_{\max} \text{ then } v_i \leftarrow \frac{v_i v_{\max}}{|v_i|}$$

$$v_i(j) \leftarrow \begin{cases} v_i(j) & \text{if} \quad |v_i(j)| \leq v_{\max}(j) \\ v_{\max}(j)\,\text{sign}(v_i(j)) & \text{if} \quad |v_i(j)| > v_{\max}(j) \end{cases} \qquad \text{for } j \in [1, n].$$

# Inertia Weighting

- **Decreasing inertia** during the optimization process may provide better performance. Where **w** is the **Inertia Weighting**

$$v_i(k) \leftarrow w v_i(k) + \phi_1(k)(b_i(k) - x_i(k)) + \phi_2(h_i(k) - x_i(k))$$

# The Constriction Coefficient

- Inertia weighting is often implemented with a constriction coefficient. This implementation, which accomplishes the same thing as the inertia weight, where **K** is called the **constriction coefficient.**

$$v_i \leftarrow K\left[v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)\right]$$

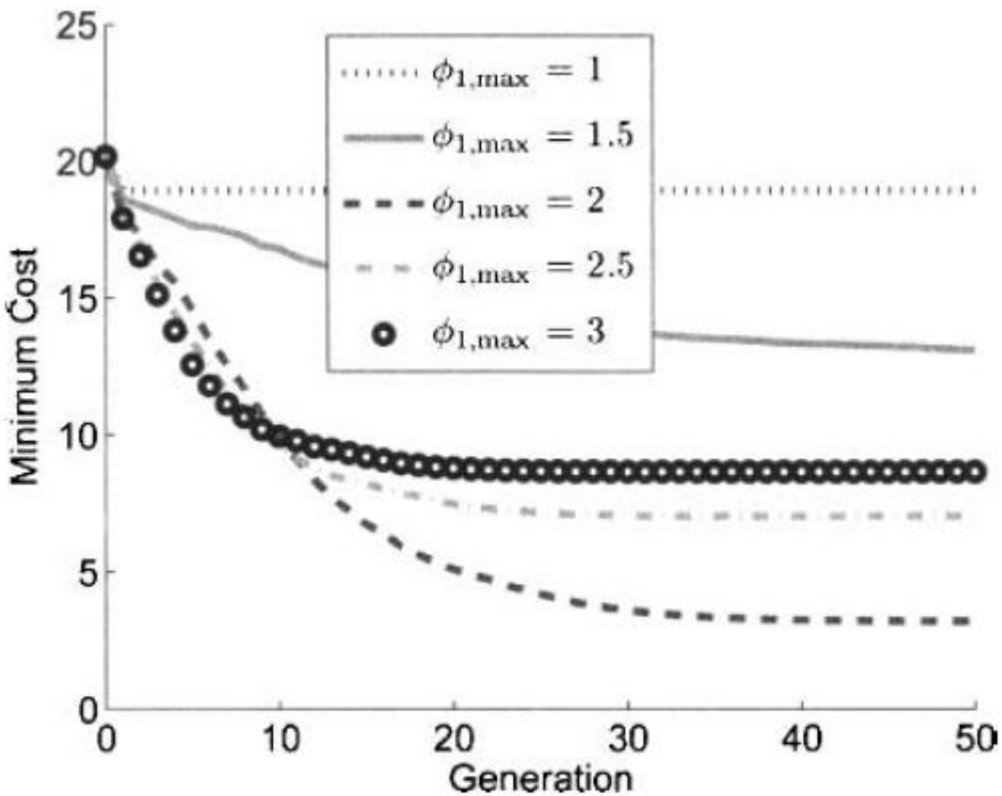
# GLOBAL VELOCITY UPDATES

- One way that we can generalize the velocity update of Equation is to write as follows.

$$v_i \leftarrow K\left[v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i) + \phi_3(g - x_i)\right]$$

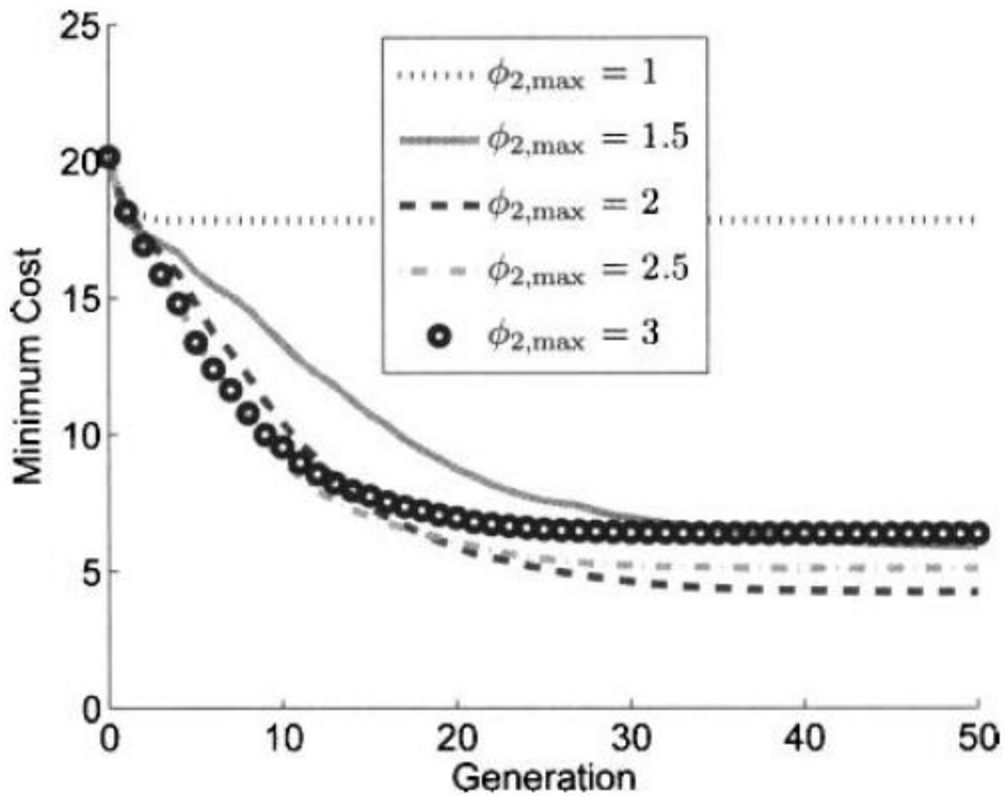
where *g is the best individual found so far since the first generation.*

# Performance



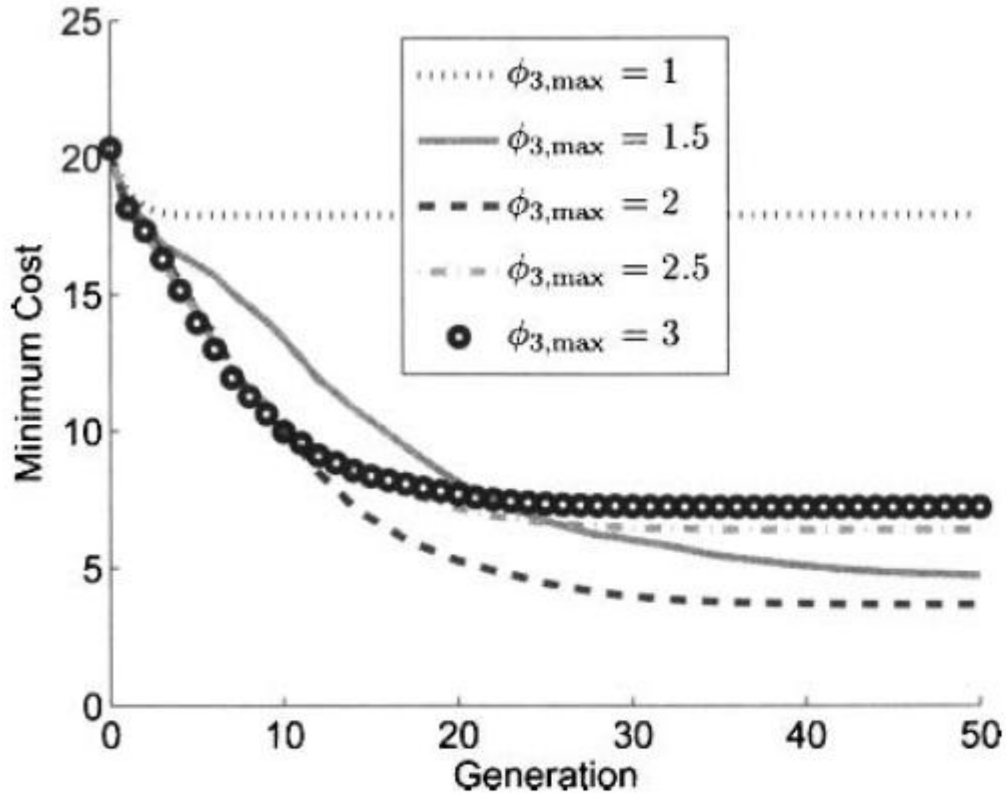

Performance of PSO on the 20-dimensional Ackley function for various values of $\phi$1,max, averaged over 20 Monte Carlo simulations. $\phi$1,max=2 approximately optimal for this benchmark function.

# Performance




Performance of PSO on the 20-dimensional Ackley function for various values of $\phi_{2,max}$, averaged over 20 Monte Carlo simulations. $\phi_{2,max}=2$ approximately optimal for this benchmark function.

# Performance




Performance of PSO on the 20-dimensional Ackley function for various values of $\phi_{3,max}$, averaged over 20 Monte Carlo simulations. $\phi_{3,max}=2$ approximately optimal for this benchmark function.

# LEARNING FROM MISTAKES

- However, biological organisms not only learn from successes, but also learn from mistakes. We tend to avoid strategies that have proven harmful in the past.
- That is termed as "negative reinforcement PSO" (NPSO)

$$\begin{aligned} v_i \quad \leftarrow \quad & K\left[v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i) + \phi_3(g - x_i)\right. \\ & \left. -\phi_4(\bar{b}_i - x_i) - \phi_5(\bar{h}_i - x_i) - \phi_6(\bar{g} - x_i)\right] \end{aligned}$$

# Reference


- Simon, Dan. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.