PSO

ALGORITHM

PRESENTED BY:
MACHAY HICHAM
LAHSSAIRI MOHEMAD AMIN
LAHKIM YAHYA

SUPERVISED BY :
AZIZ OUAARAB

CONTENT

INTRODUCTION:

- **Characteristics and Inspiration from Natural Systems (ants, bees, flocking birds)**

Ants:

Ants in a group are like a super team. Even though each ant is simple, they follow some rules to get things done, like finding food or building a home. They talk to each other using special smells.

INTRODUCTION:

- **Characteristics and Inspiration from Natural Systems (ants, bees, flocking birds)**

Bees:

Bees are like friends who have different jobs but together make a strong community. They have a dance language to share where the best food is, showing how they work together without a boss.

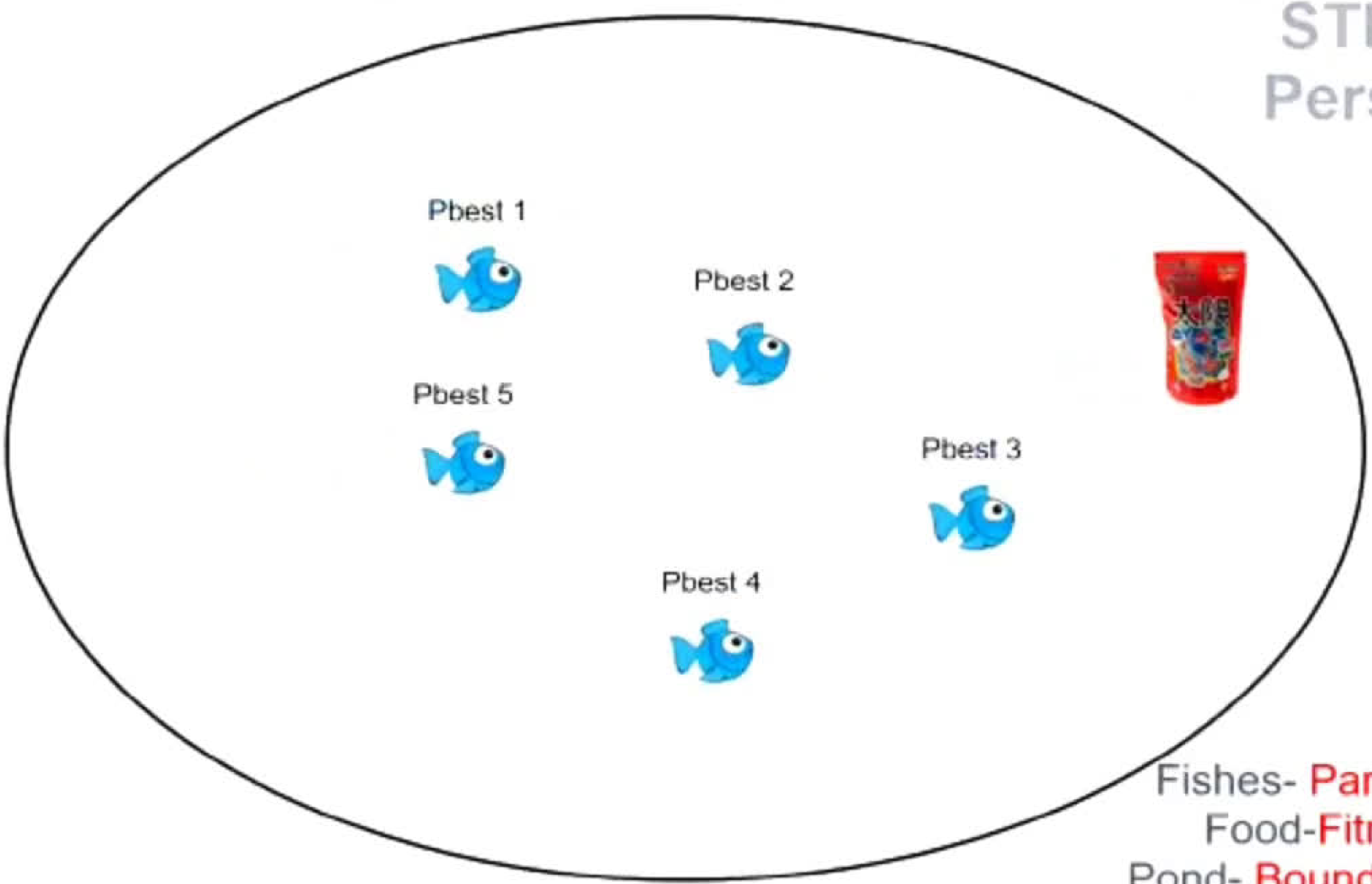
INTRODUCTION:

- **Characteristics and Inspiration from Natural Systems (ants, bees, flocking birds)**

Flocking Birds:

Picture a bunch of birds flying together, making cool shapes in the sky. They're not following a leader, but each bird follows simple rules, making them look amazing as a group.

STEP 2: Find Personal Best



Fishes- Particle/Variables
Food-Fitness/fitness
Pond- Boundary/Constraints

INTRODUCTION:

- **Definition of Swarm Intelligence**

- Swarm Intelligence is like a cool science that learns from how animals in groups work together in nature. It helps solve big and tricky problems. Imagine a team where each member isn't very smart on their own, but when they work together, they become super clever.
- In simpler terms, it's about groups of things working together without a boss telling them what to do. These things could be animals, like ants and bees, or even birds flying together in a cool pattern.

INTRODUCTION:

- **Definition of Swarm Intelligence**

  - the expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems in their paper titled "Theoretical problems in artificial life: Cellular automata as dynamical systems.".
  - Particle Swarm Optimization (PSO) is a specific application of SI that leverages these principles to efficiently explore solution spaces and find optimal solutions to complex problems.

UNDERSTANDING PARTICLE SWARM OPTIMIZATION (PSO)

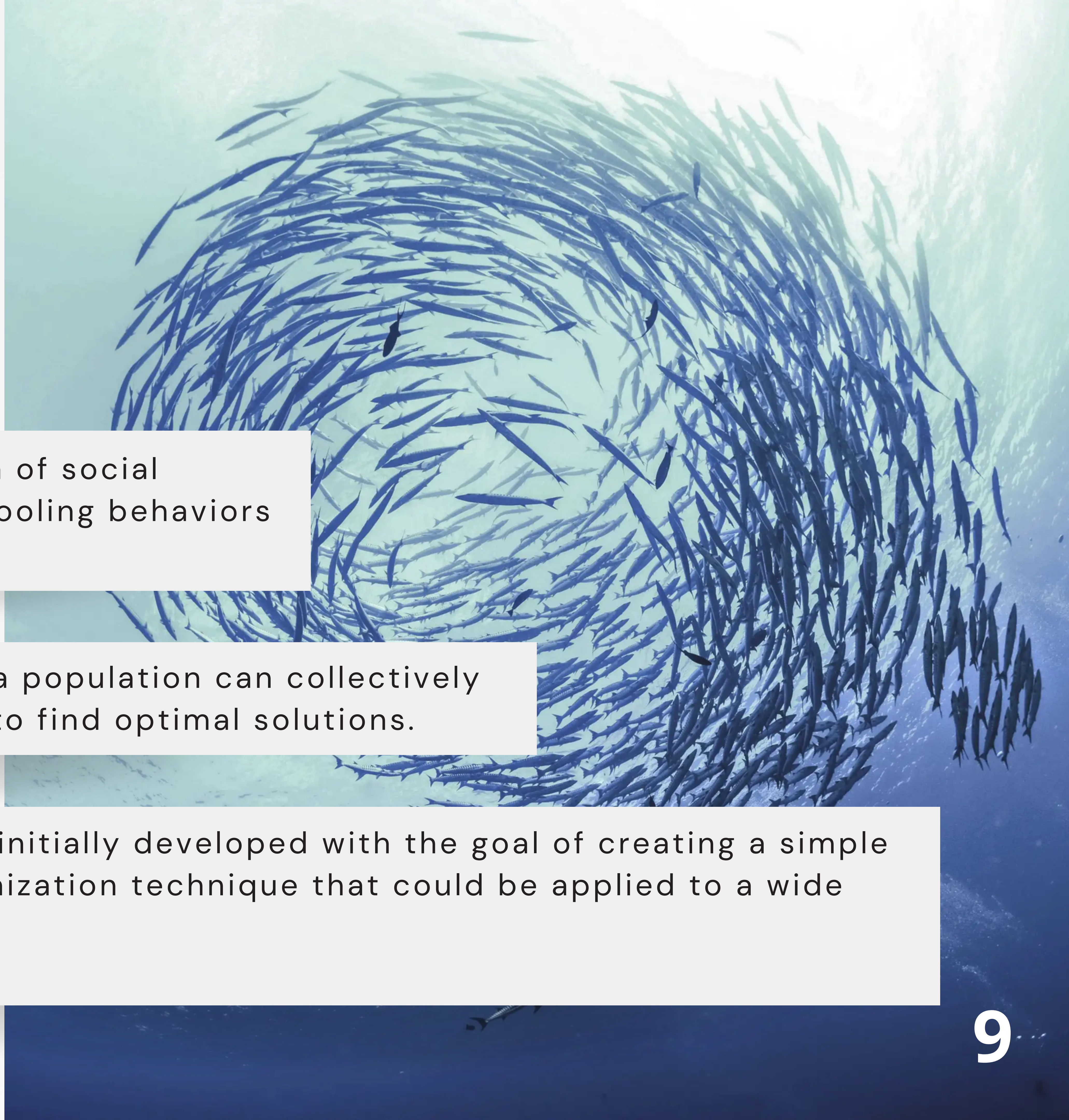
WHAT IS PSO ALGORITHM?

Particle Swarm Optimization (PSO) is a metaheuristic inspired by the social behavior of bird flocks or fish schools. It was first introduced by Kennedy and Eberhart in 1995, and since then it has become a popular optimization algorithm for a wide range of problems.

In PSO, a group of particles (representing potential solutions) navigates through a problem's solution space to find the best possible solution.

Each particle adjusts its position based on its own best-known solution (personal best) and the best solution discovered by the entire group (global best). This collaborative movement enables particles to converge toward optimal solutions over iterations.

ORIGIN OF PSO:

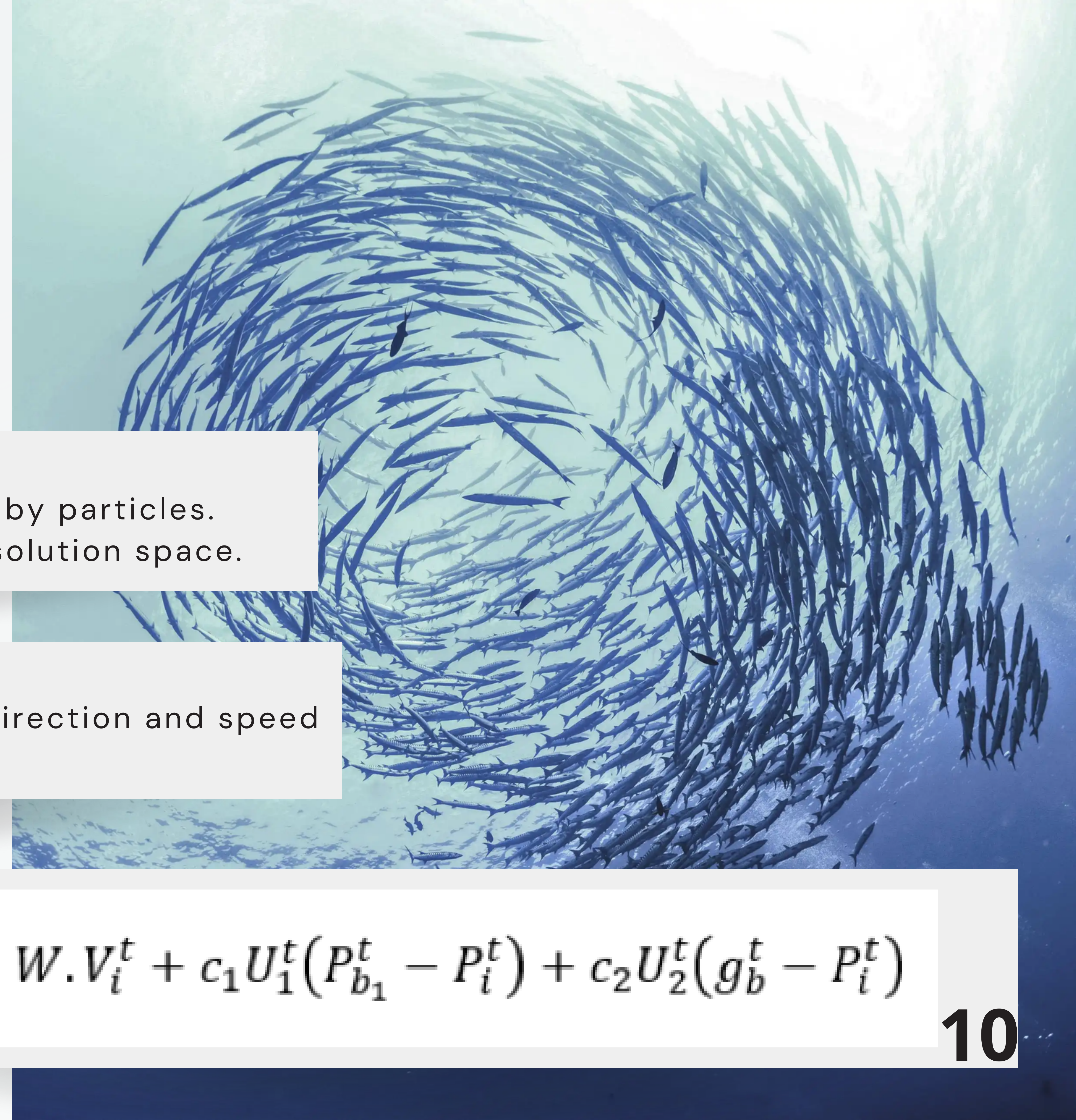


The concept of PSO originated from the observation of social behavior in nature, particularly the flocking and schooling behaviors of birds and fish.

The idea is based on the premise that individuals in a population can collectively and intelligently explore and exploit a search space to find optimal solutions.

The algorithm was initially developed with the goal of creating a simple yet effective optimization technique that could be applied to a wide range of problems.

KEY COMPONENTS OF PSO:



1- Particles:

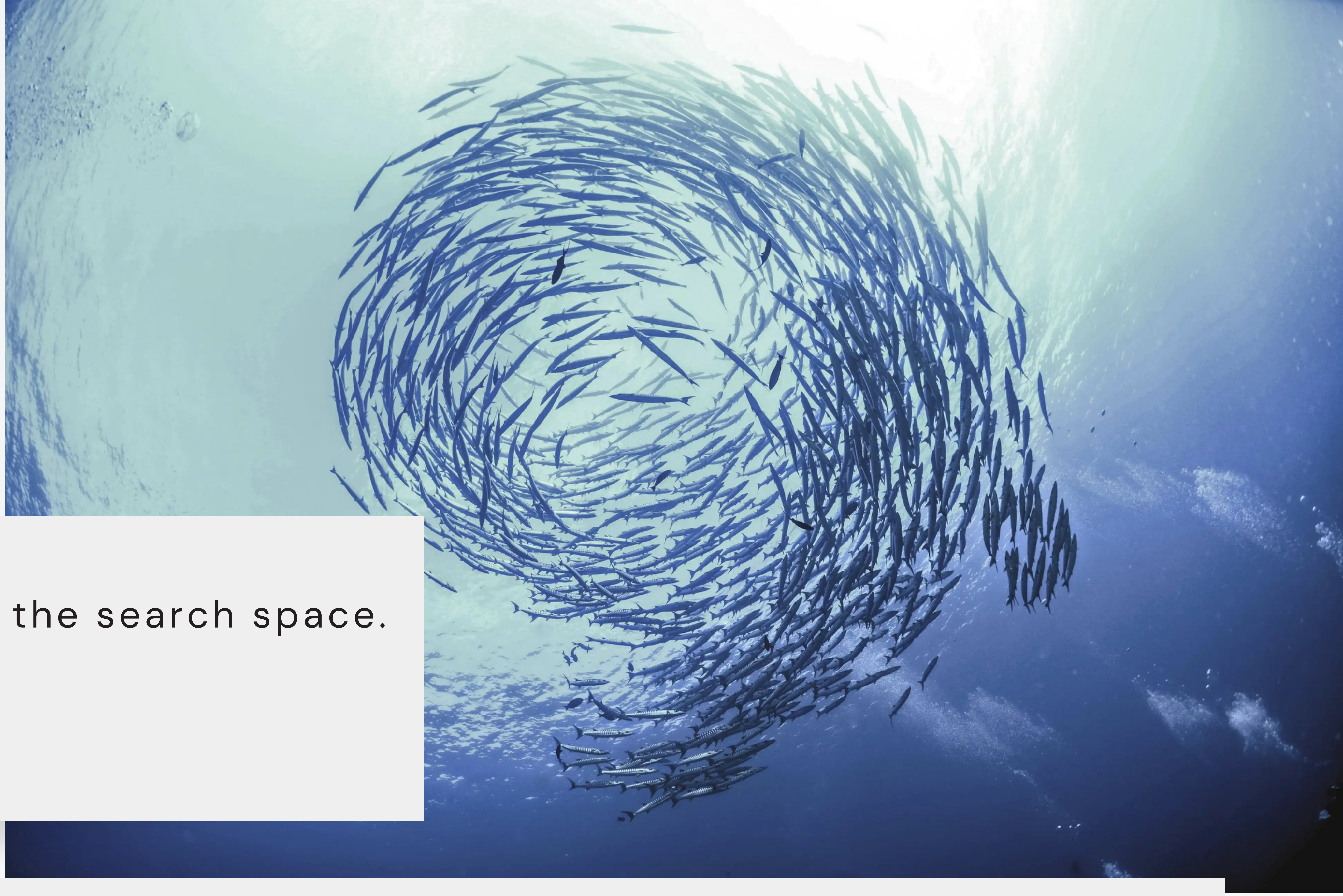
- A population of potential solutions, represented by particles.
- Each particle has a position and velocity in the solution space.

2- Velocity Vi:

- Each particle has a velocity, which indicates its direction and speed of movement in the search space.

$$V_i^{t+1} = W.V_i^t + c_1 U_1^t\left(P_{b_1}^t - P_i^t\right) + c_2 U_2^t\left(g_b^t - P_i^t\right)$$

KEY COMPONENTS OF PSO:



3- Position:

- The current solution represented by a particle in the search space.

$$P_i^{t+1} = P_i^t + v_i^{t+1}$$

4- Personal Best (Pbest):

- Each particle maintains its own personal best position, which is the best position it has found so far.

5- Global Best (gBest):

- The global best position is the best position found by any particle in the swarm.

KEY COMPONENTS OF PSO:

6- Inertia Weight (W):

- Parameter that influences the particle's tendency to continue in its current direction.
- It controls the trade-off between exploration and exploitation.

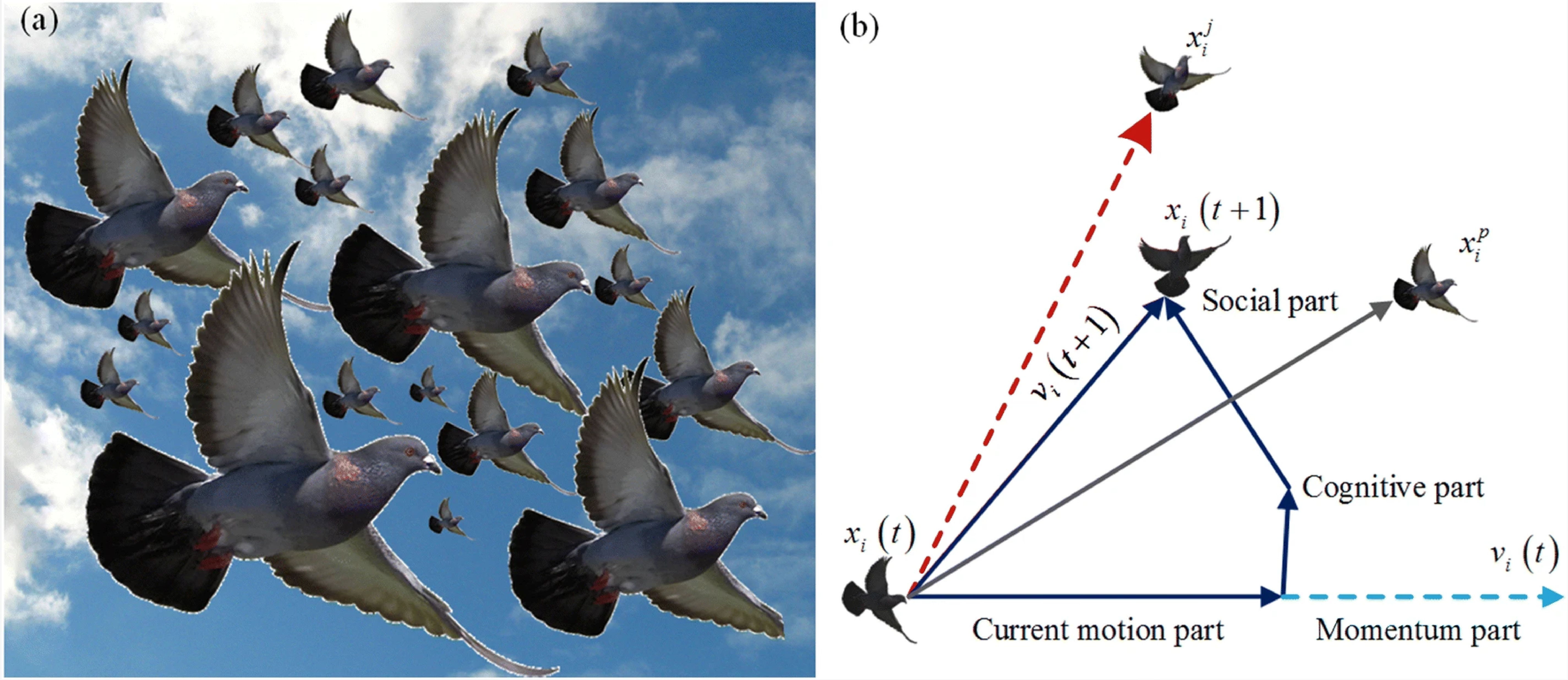
7- Cognitive and Social Components:

- Cognitive Component (c1): A parameter representing the particle's tendency to follow its personal best.
- Social Component (c2): A parameter representing the particle's tendency to follow the global best.

KEY COMPONENTS OF PSO:

- **Fitness Function**
- A fitness function evaluates the quality of a particle's current position (solution).
- It provides a measure of how well the solution satisfies the optimization problem's objective.

MECANISME OF PSO:



MATHEMATICAL FORMULATION OF PSO

$$V_i^{t+1} = W.V_i^t + c_1 U_1^t (P_{b_1}^t - P_i^t) + c_2 U_2^t (g_b^t - P_i^t)$$

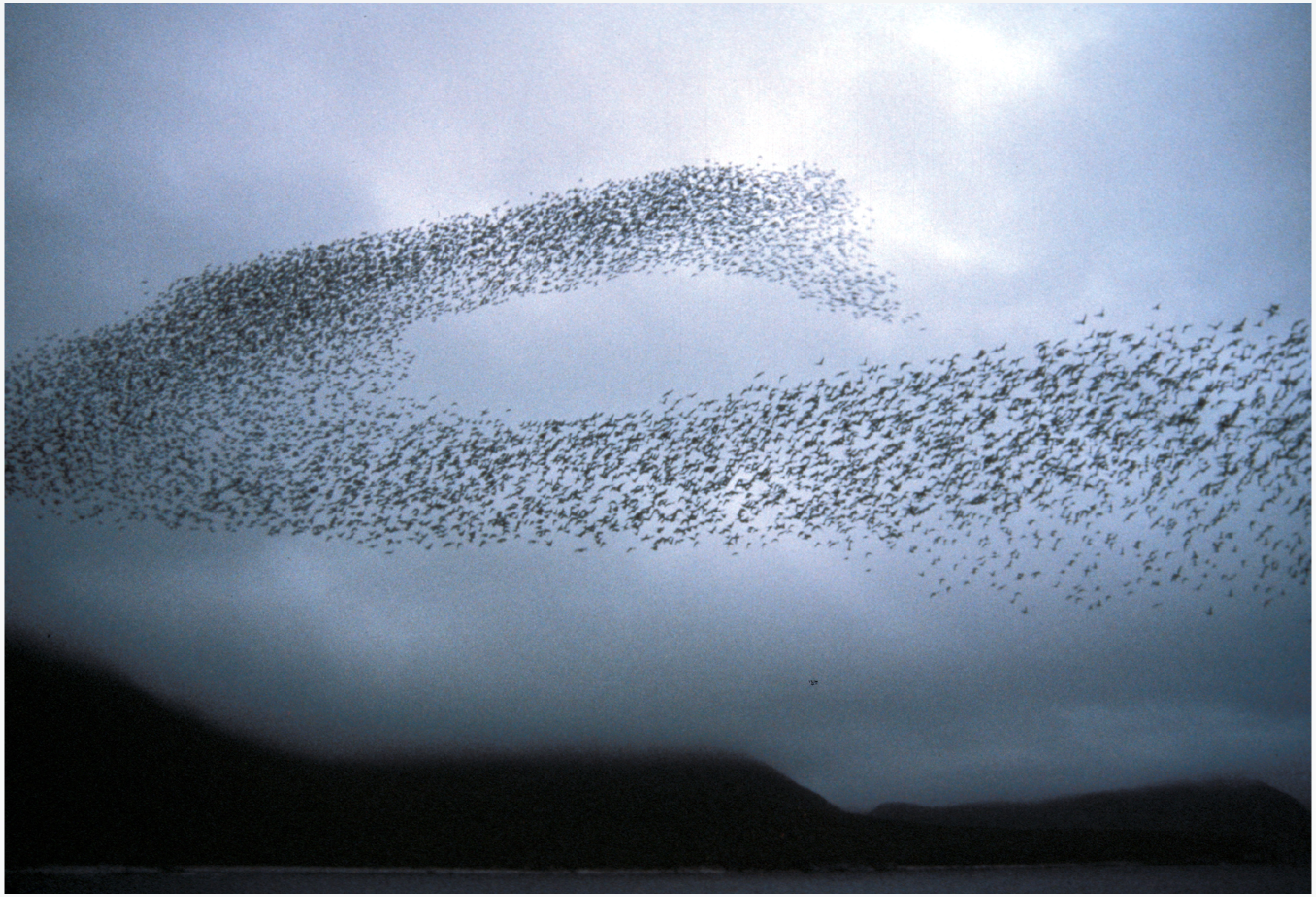
Personal Influence: Makes the particle return to a previous position, better than the current.

Social Influence: Makes the particle follow the best neighbors direction.

$$P_i^{t+1} = P_i^t + v_i^{t+1}$$

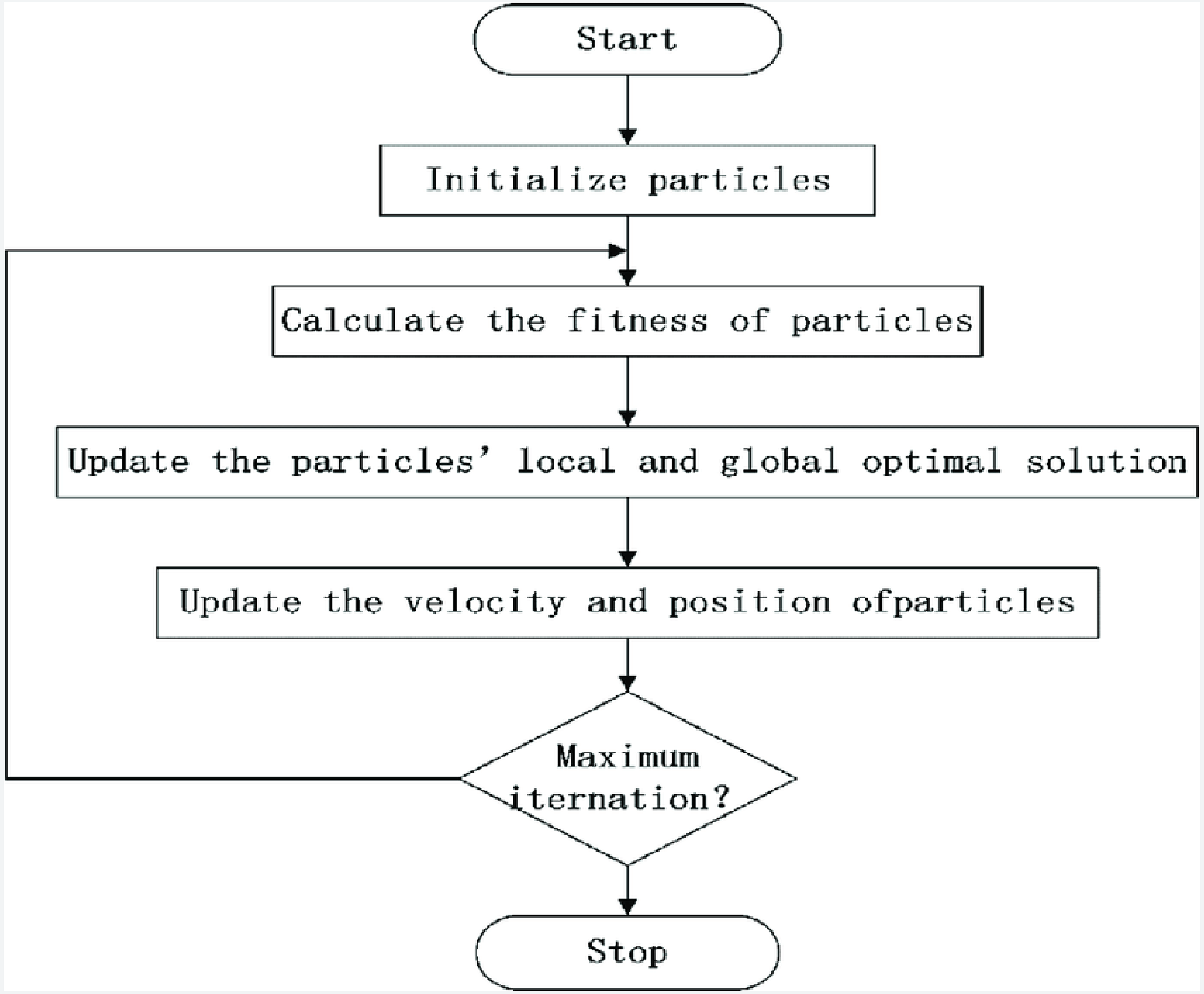
NEIGHBORHOOD

Imagine a swarm of birds flying together. Each bird is aware of the positions of nearby birds, and this information guides its own movement. In PSO, the neighborhood topology defines which nearby birds a particle can interact with. This interaction influences the particle's velocity update, guiding it towards better solutions.

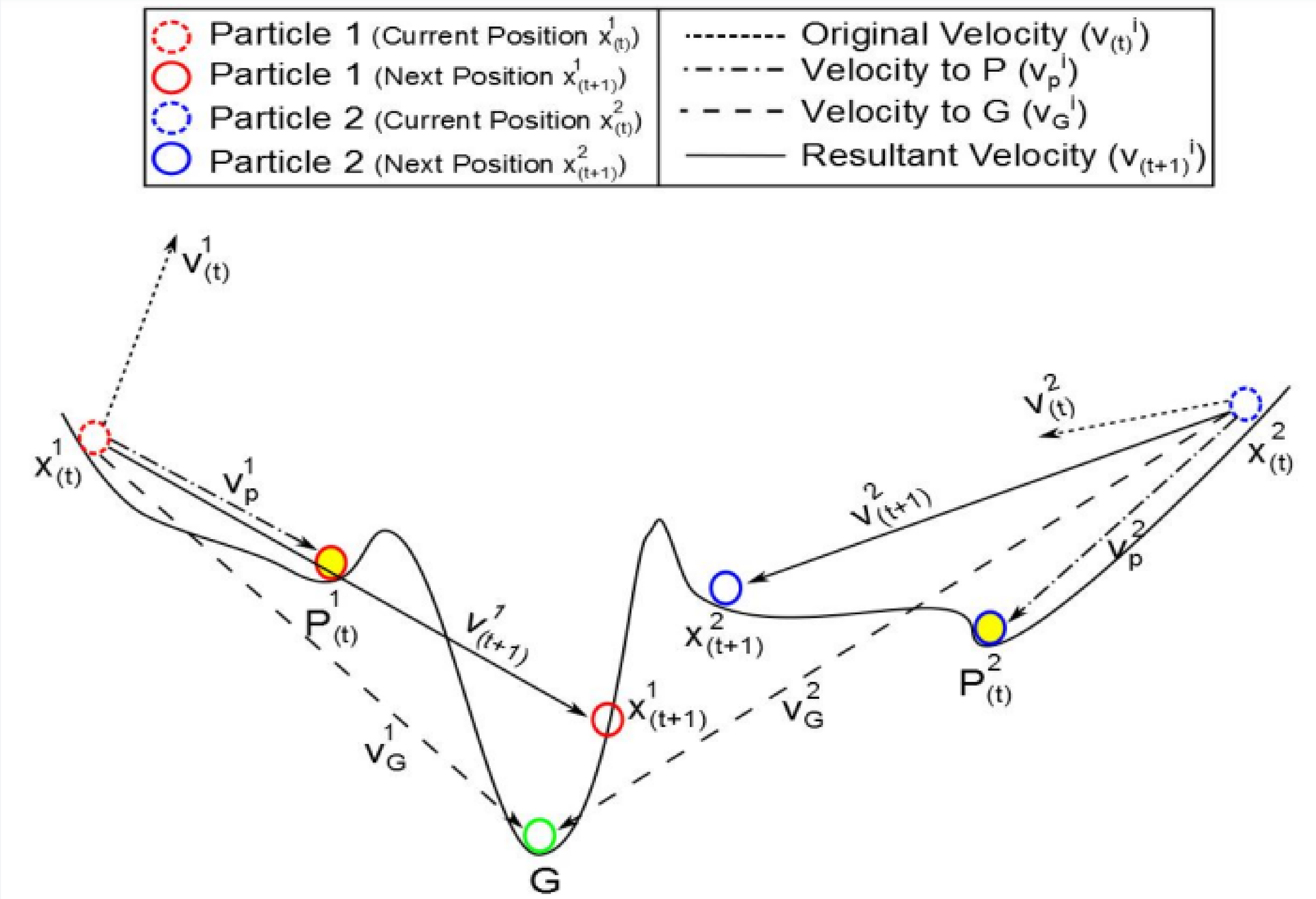


PSO ALGORITHM

WORKFLOW OF PSO:



EXAMPLE :



Particle 1 (Current Position $x^1_{(t)}$)
Particle 1 (Next Position $x^1_{(t+1)}$)
Particle 2 (Current Position $x^2_{(t)}$)
Particle 2 (Next Position $x^2_{(t+1)}$)
Original Velocity ($v_{(t)}{}^i$)
Velocity to P ($v_p{}^i$)
Velocity to G ($v_G{}^i$)
Resultant Velocity ($v_{(t+1)}{}^i$)
$v^1_{(t)}$
$x^1_{(t)}$
v^1_p
$P^1_{(t)}$
$v^1_{(t+1)}$
$x^1_{(t+1)}$
v^1_G
G
$v^2_{(t)}$
$x^2_{(t)}$
$v^2_{(t+1)}$
v^2_p
$x^2_{(t+1)}$
$P^2_{(t)}$
v^2_G

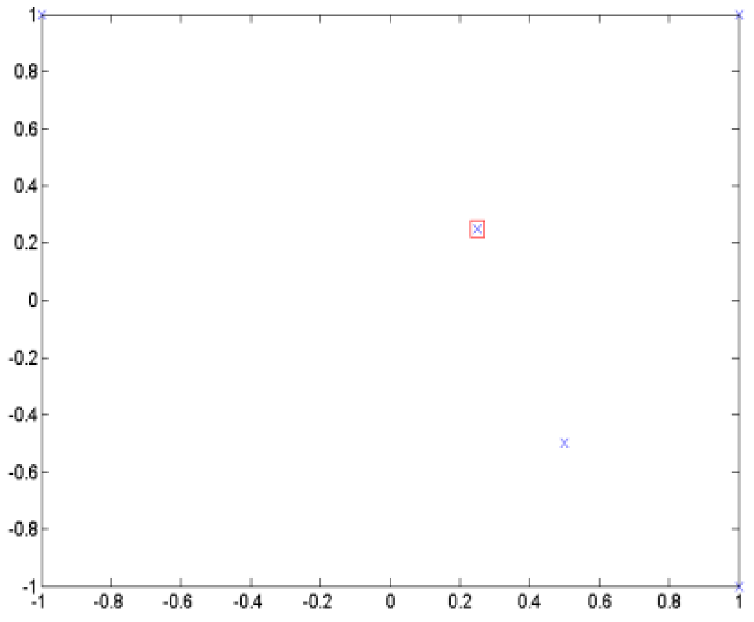
EXAMPLE :

Particle No.	Initial Positions		Velocity		Best Solution	Best Position		Fitness Value
	x	y	x	y		x	y	
P_1	1	1	0	0	1000	-	-	2
P_2	-1	1	0	0	1000	-	-	2
P_3	0.5	-0.5	0	0	1000	-	-	0.5
P_4	1	-1	0	0	1000	-	-	2
P_5	0.25	0.25	0	0	1000	-	-	0.125

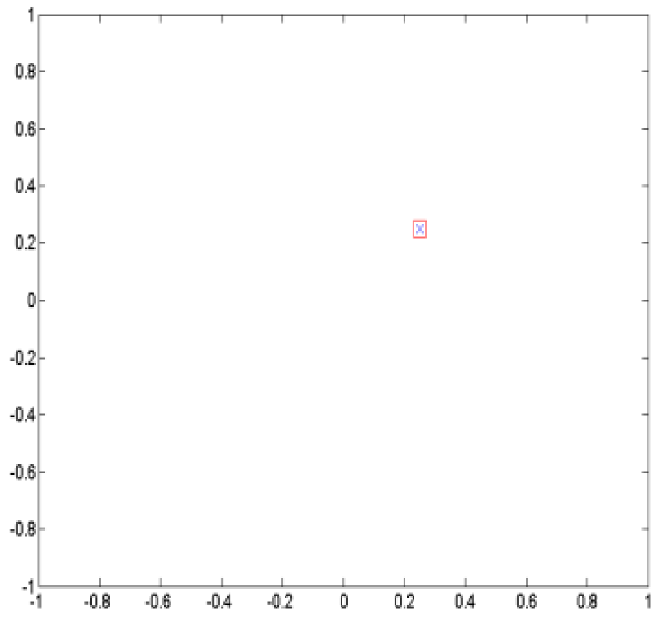
First iteration

Particle No.	Initial Positions		Velocity		Best Solution	Best Position		Fitness Value
	x	y	x	y		x	y	
P_1	1	1	-0.75	-0.75	2	1	1	2
P_2	-1	1	1.25	-0.75	2	-1	1	2
P_3	0.5	-0.5	-0.25	0.75	0.5	0.5	-0.5	0.5
P_4	1	-1	-0.75	1.25	2	1	-1	2
P_5	0.25	0.25	0	0	0.125	0.25	0.25	0.125

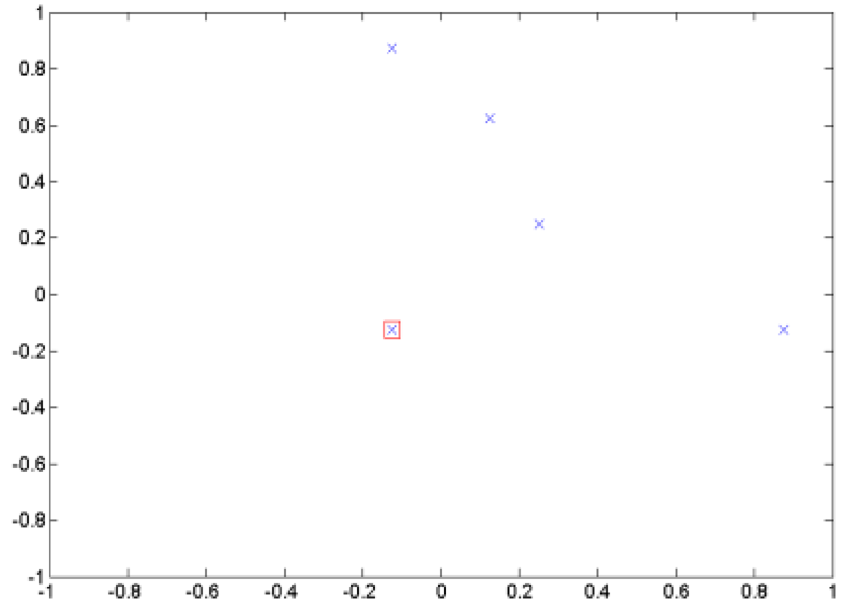
EXAMPLE :



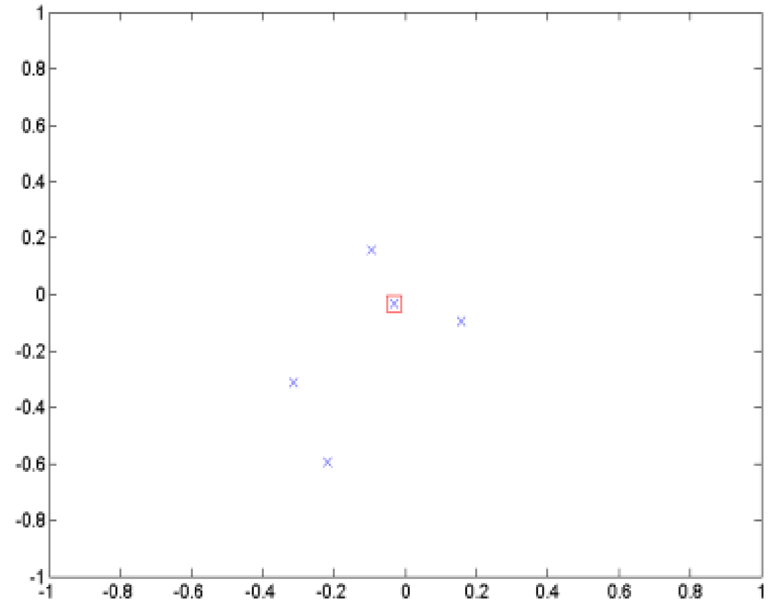
(a) First Iteration



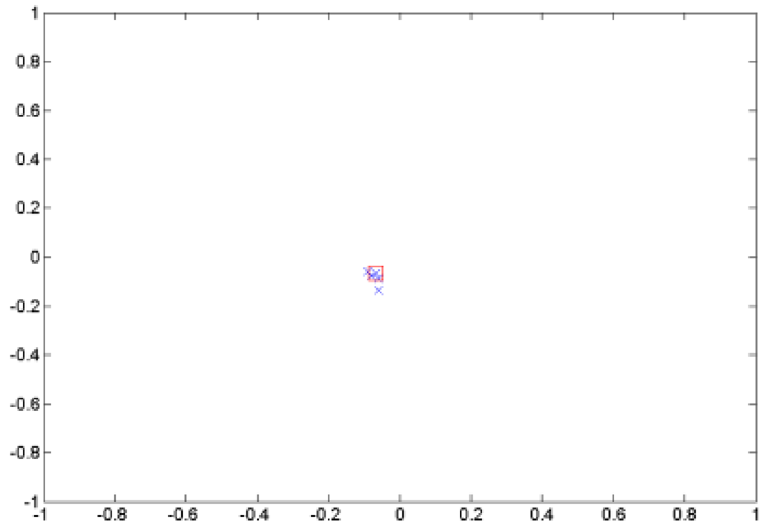
(b) Second Iteration



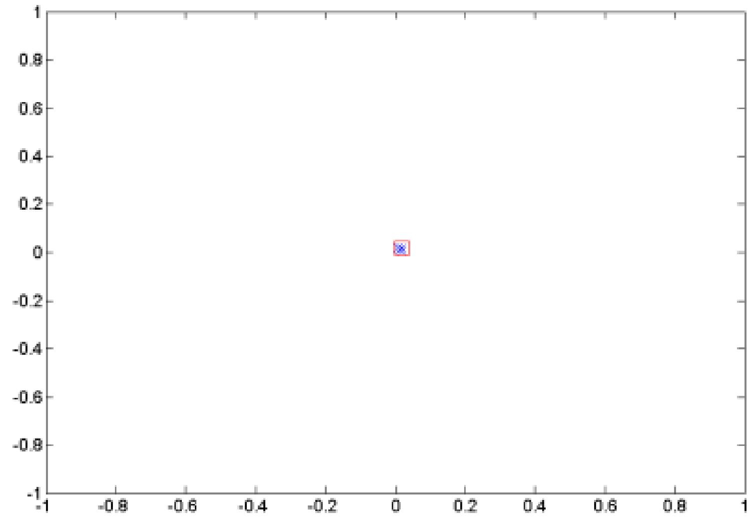
(c) Third Iteration



(d) Fourth Iteration



(e) Seventh Iteration



(f) Thirteenth Iteration

DIVERSIFICATION AND INTENSIFICATION

DIVERSIFICATION AND INTENSIFICATION

Diversification refers to the exploration of the search space to discover new and unexplored regions. It prevents the algorithm from getting trapped in local optima by encouraging particles to search in different areas, potentially finding better solutions.

Intensification focuses on exploiting known promising regions in the search space. It involves refining and exploiting the best solutions found so far, aiming to converge towards the optimal solution.

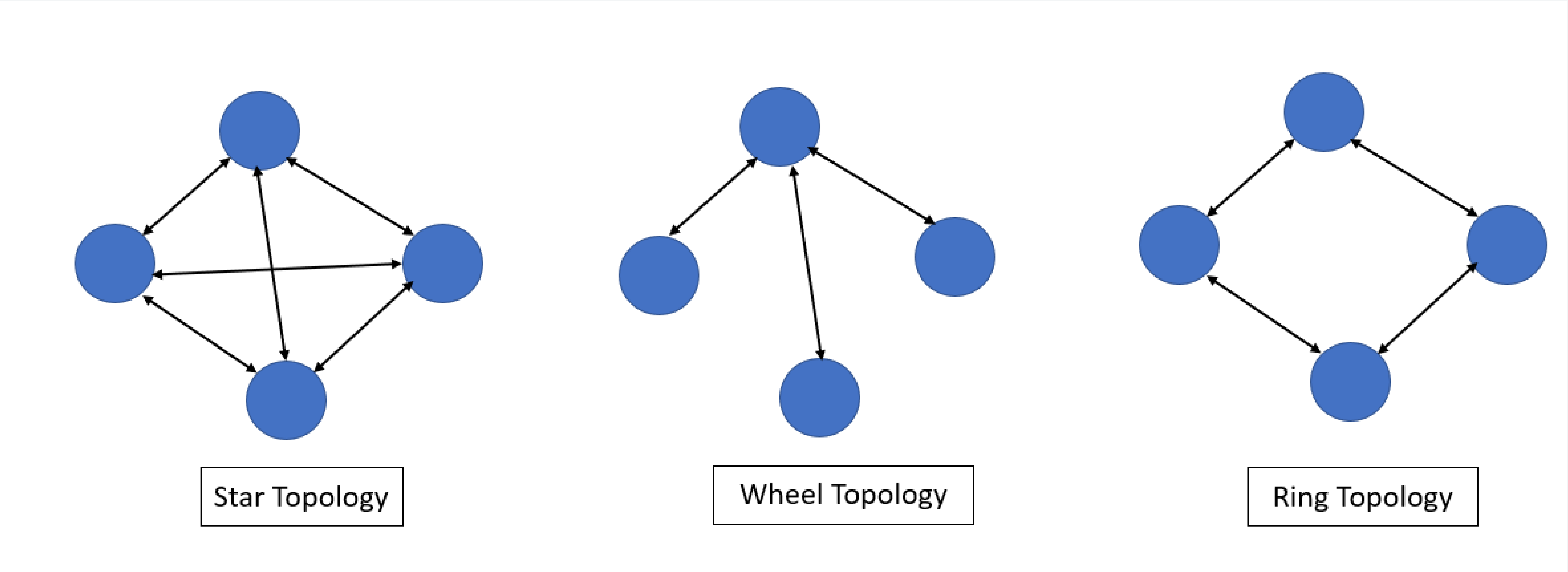
DIVERSIFICATION AND INTENSIFICATION

Inertia Weight (W) : When the inertia weight is set high, particles tend to maintain their velocity from the previous iterations more prominently. This promotes exploration by allowing particles to move across a wider search space, potentially discovering new regions and avoiding getting stuck in local optima.

C1 and C2 : Balancing the values of C1 and C2 is crucial to strike a proper trade-off between exploration and exploitation in PSO.
If the values of C1 and C2 are too high, the algorithm might focus excessively on exploiting known solutions, potentially missing out on exploring new regions of the search space.

SWARM TOPOLOGY

NEIGHBORHOOD TOPOLOGIES



NEIGHBORHOOD TOPOLOGIES

- **Definition:**

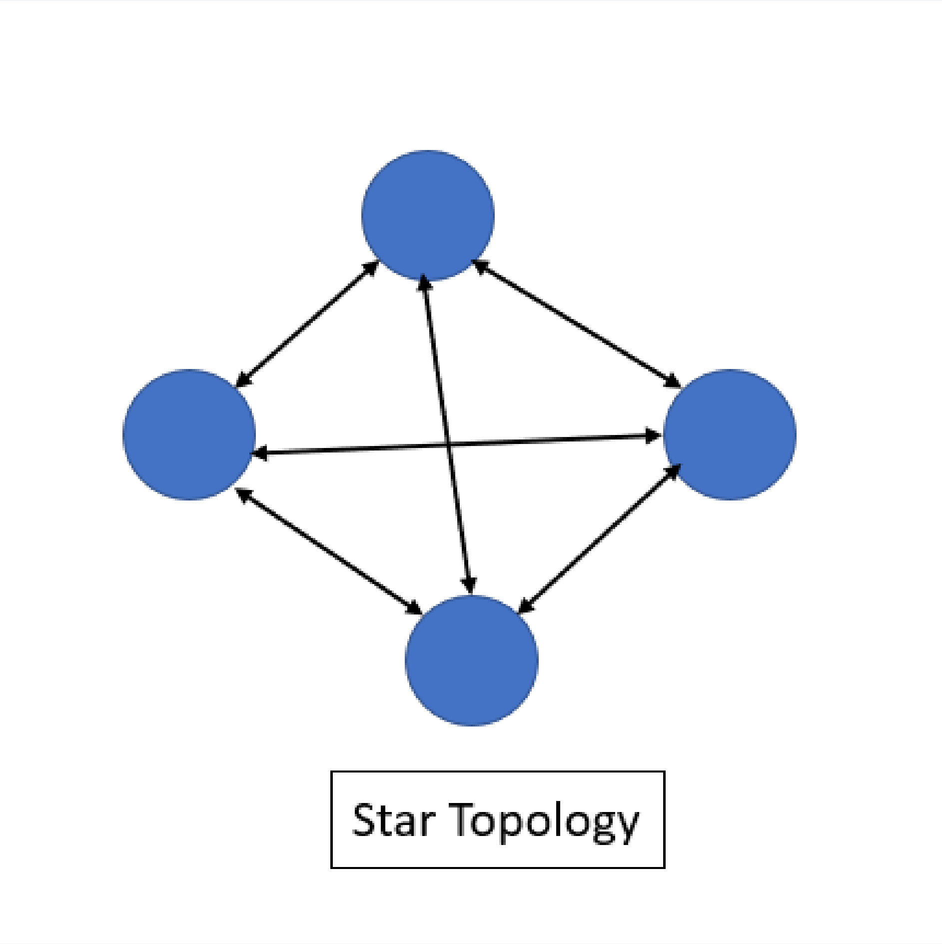
In a star topology, each particle communicates directly with a central particle, often referred to as the global best or leader.

- **Explanation:**

Think of it as a star with all particles connected to a central point. Each particle is aware of the global best solution, and their movements are influenced by this central point.

- **Goal in PSO:**

The central particle represents the best solution found so far. Each individual particle adjusts its position based on its own experience (personal best) and the overall best (global best) found by the central particle.



NEIGHBORHOOD TOPOLOGIES

- **Definition:**

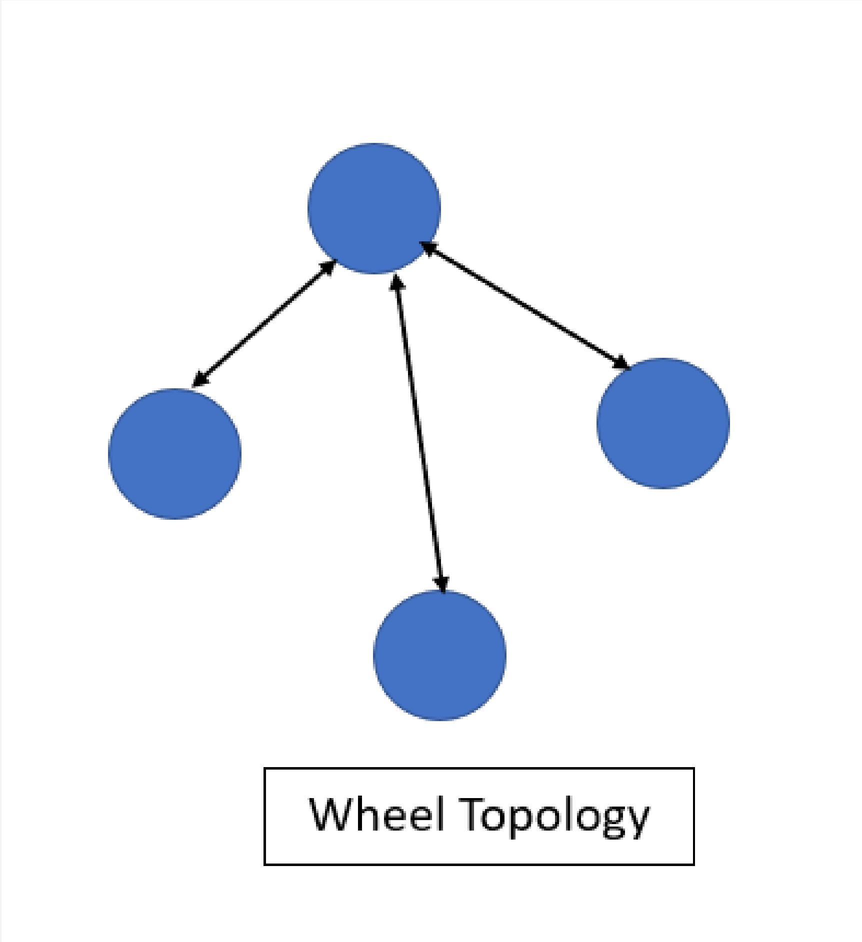
In a wheel topology, particles are connected in a circular manner, forming a structure similar to the spokes of a wheel.

- **Explanation:**

Imagine a wheel where each particle is connected to two neighboring particles. The last particle is connected to the first, creating a circular structure.

- **Goal in PSO:**

Similar to star topology, each particle knows about the global best. However, instead of a single central particle, information flows in a circular manner. Each particle is influenced by its neighbors, and this circular connection helps in sharing information efficiently.



NEIGHBORHOOD TOPOLOGIES

- **Definition:**

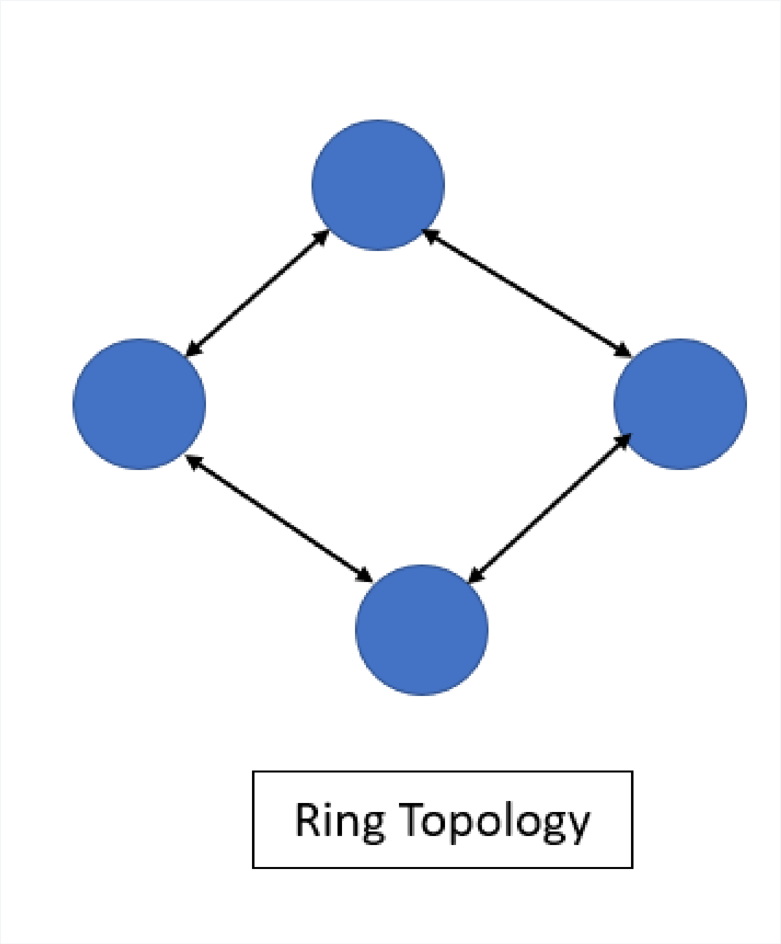
In a ring topology, particles are connected in a linear or circular fashion, forming a chain or loop.

- **Explanation:**

Visualize particles arranged in a line or a circle, with each particle connected to its adjacent ones.

- **Goal in PSO:**

Like the wheel topology, particles in a ring topology share information with their neighbors. The last particle is connected to the first, forming a closed loop. This arrangement facilitates the flow of information among particles in a circular manner.



VARIANTS OF PSO

BINARY PSO (BPSO)

Binary PSO (BPSO) is an extension of the original PSO algorithm designed to handle optimization problems with binary variables. These problems often arise in feature selection, rule generation, and discrete optimization. BPSO utilizes a modified velocity update rule that converts continuous velocities to binary values, ensuring that the particle positions remain within the binary domain.

The velocity update rule in BPSO is typically based on a sigmoid function, which maps continuous values to probabilities between 0 and 1. These probabilities represent the likelihood of flipping each bit in the binary representation of the particle's position. The update rule then applies random number comparisons to determine whether each bit should actually be flipped.

BPSO has been successfully applied to various binary optimization problems, demonstrating its effectiveness in handling discrete decision variables and achieving optimal solutions.

ADAPTIVE PSO

Adaptive PSO dynamically adjusts PSO parameters, such as inertia weight, cognitive coefficient, and social coefficient, based on optimization progress or problem characteristics. This adaptive approach enhances the algorithm's performance for diverse optimization problems.

By monitoring the optimization progress and the behavior of particles, adaptive PSO can identify when to adjust the parameters to promote either exploration or exploitation. For instance, if the swarm converges prematurely, the inertia weight can be increased to encourage further exploration. Conversely, if the swarm becomes stagnant, the cognitive and social coefficients can be increased to focus the search around promising regions.

Adaptive PSO has demonstrated significant improvements in optimization performance compared to traditional PSO with fixed parameters. The ability to adapt to the problem dynamics and the behavior of the swarm makes adaptive PSO a powerful tool for a wide range of optimization challenges.

QUANTUM-BEHAVED PSO (QPSO)

Quantum-Behaved Particle Swarm Optimization (QPSO) applies quantum mechanics principles to enhance the traditional PSO algorithm. By introducing a quantum-inspired update rule and a quantum rotation gate, QPSO enables particles to explore the solution space more efficiently. This variant facilitates improved exploration of uncharted regions while maintaining diversity among particles, aiding in preventing premature convergence. QPSO's unique quantum behavior enhances the algorithm's convergence speed, making it suitable for optimization challenges where efficient exploration is crucial, such as in machine learning, feature selection, and data clustering applications.

ADVANTAGES AND CHALLENGES OF PSO

ADVANTAGES OF PSO:

- 1. Insensitive to scaling of design variables.
- 2. Easily parallelized for concurrent processing.
- 3. Derivative free.
- 4. Very few algorithm parameters.
- 5. A very efficient global search algorithm.

DISADVANTAGE OF PSO:

- PSO’s optimum local searchability is weak

DIFFERENCE BETWEEN PSO & GENETIC ALGORITHMS

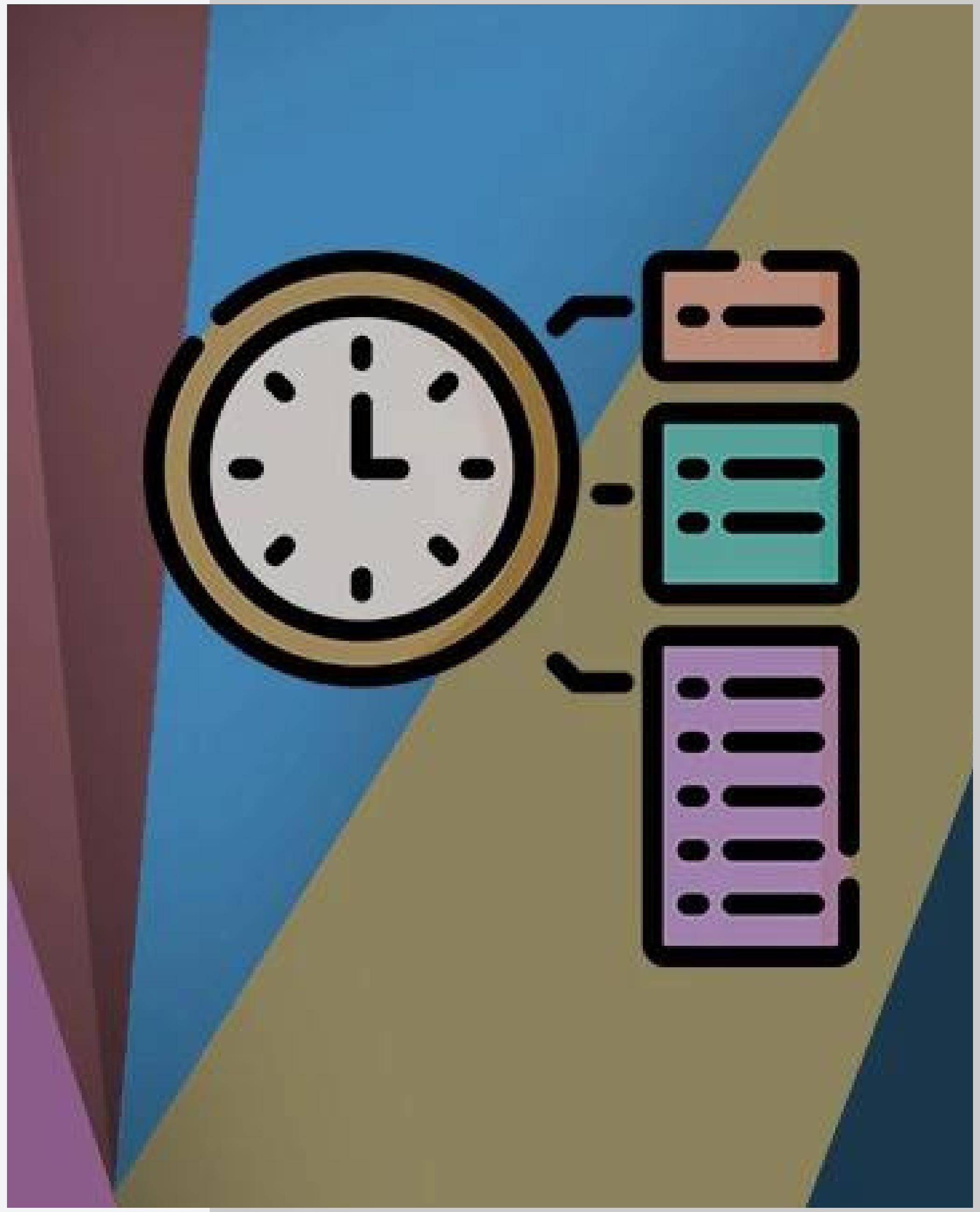
GAs	PSO
GAs are inspired by the process of biological evolution.	PSO is inspired by the social behaviour of bird flocks or fish schools.
GAs use genetic operators to create new candidate solutions.	PSO uses particle positions and velocities to update the swarm's position.
GAs can converge to a local optimum.	PSO is less prone to getting stuck in local optima.
GAs create new individuals through reproduction.	PSO does not create or delete individuals.
GAs do not allow individuals to communicate with each other.	PSO algorithms allow particles to communicate with each other.

APPLICATIONS OF PSO

APPLICATIONS OF PSO:

Job Scheduling:

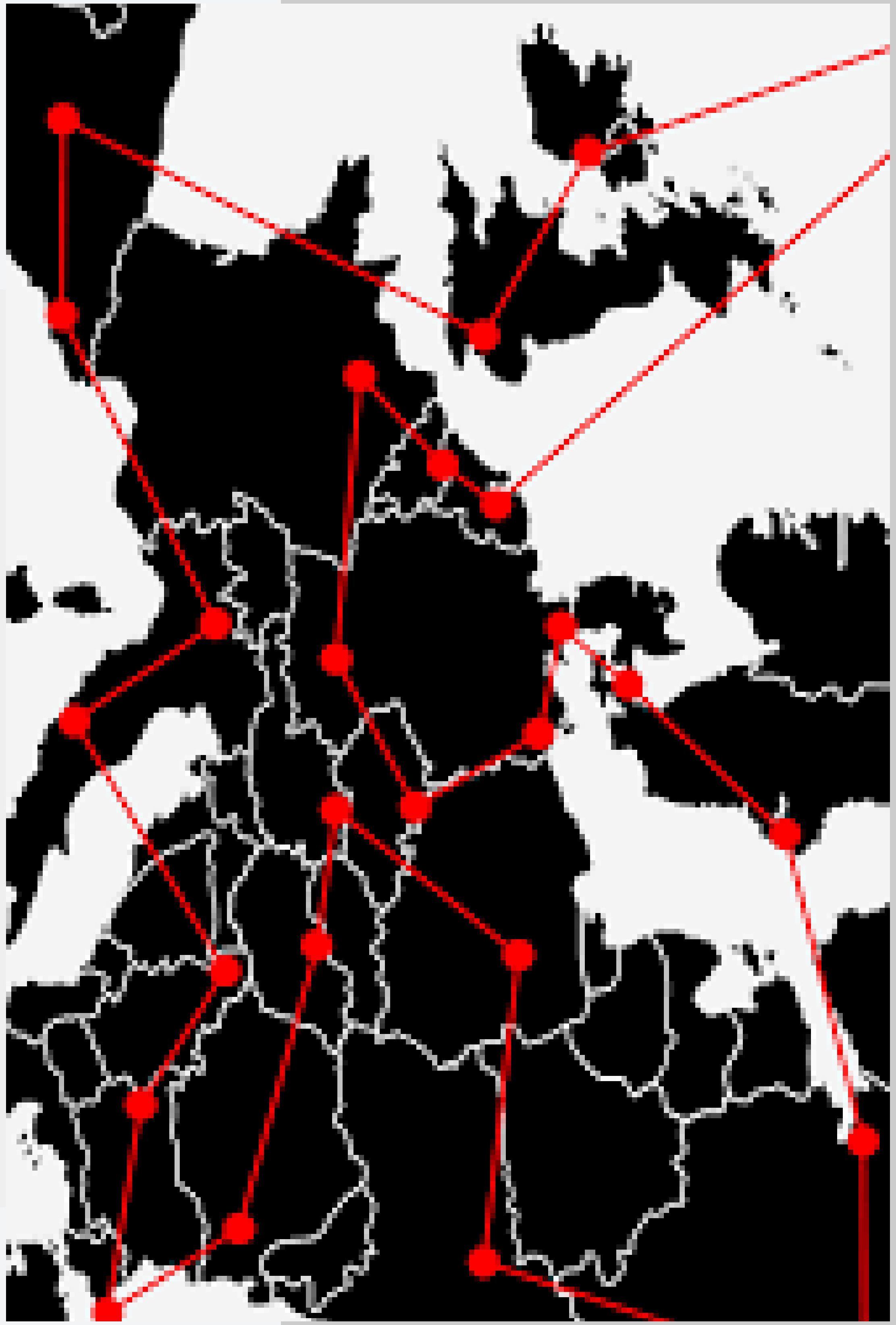
PSO is applied in optimizing job scheduling problems, helping allocate resources efficiently and minimize makespan in various scheduling scenarios.



APPLICATIONS OF PSO:

Traveling Salesman Problem (TSP):

PSO is effective in solving the Traveling Salesman Problem, finding optimal or near-optimal routes for visiting a set of cities and returning to the starting city.



Conclusion

Thank you for your attention

MCQ PSO :

1. What is Particle Swarm Optimization (PSO)?
(a) A metaheuristic inspired by the social behavior of bird flocks or fish schools
(b) A gradient-based optimization algorithm
(c) A local search algorithm
(d) A global search algorithm

2. What is the role of inertia weight in PSO?
(a) It controls the balance between exploration and exploitation
(b) It determines the step size of the particles' movements
(c) It prevents the particles from getting stuck in local optima
(d) It ensures that the particles converge to the global optimum

3. In PSO, what is the role of velocity in updating particle positions?
a) It represents the fitness of a particle
b) It determines the direction and speed of movement
c) It is used as a randomization factor
d) It is irrelevant in PSO

```
Initialize Xi, Vi, iteration, pbest, gbest
Generate random particles (P)
For each particle (i)
   Calculate fitness function (fi)
   Update pbest, gbest
End for
While iteration
   For each particle I
      Update Vi, Xi
      If Xi > limit then Xi = limit
      Calculate fitness function fi
      Update pbest, gbest
   End for
End while
```