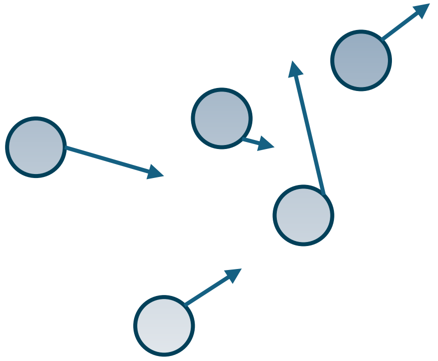
Lecture 8

- **Particle Swarm Optimization (PSO) Algorithm**
- **PSO Java implementation**

CENG 632- Computational Intelligence, 2024-2025, Spring

Assist. Prof. Dr. Osman GÖKALP

Particle Swarm Optimization



- The Particle Swarm Optimization (PSO) algorithm was proposed by Kennedy and Eberhart in 1995.
- It is among the most successful and important algorithms used for **numerical (continuous) optimization**.
  - See the presentation “Lecture 3” for a review of numerical optimization problems.
- **Key features**
  - Population-based
  - Maintains velocity vectors
  - Uses both particles’ best and global best positions

PSO solution representation

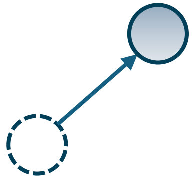
Since **PSO** is a **numerical optimization** algorithm, the solutions are in the form of a **vector of real numbers**.

$$\vec{X} = (x_1, x_2, \dots, x_D),\ where\ x_i \in [a_i, b_i],\ \forall i = 1,2, \dots, D.$$

x_1	x_2	**…**	x_D

The **solution** $\vec{X}$ consists of D **real numbers**. In other words, the problem is D dimensional.

Particles



- A **Particle** is a building block of a PSO population.
- Each particle maintains the following 3 vectors:
  - **Velocity** vector

    $\vec{V} = (v_1, v_2, \dots, v_D),\ v_i \in [-|u_i - l_i|, |u_i - l_i|],\ \forall i = 1,2, \dots, D$

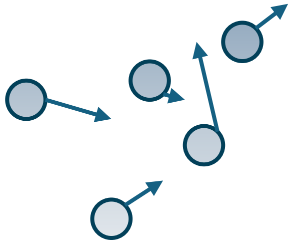
  - **Current position**, or the solution vector

    $\vec{X} = (x_1, x_2, \dots, x_D),\ x_i \in \left[l_{i,} u_i\right],\ \forall i = 1,2, \dots, D$

  - Its **best known position** vector

    $\vec{P} = (p_1, p_2, \dots, p_D),\ p_i \in \left[l_{i,} u_i\right],\ \forall i = 1,2, \dots, D$

Populations



- A PSO **population** consists of multiple particles:

$$P = \{(\vec{X}_1, \vec{V}_1, \vec{P}_1), (\vec{X}_2, \vec{V}_2, \vec{P}_2), \ldots, (\vec{X}_{NP}, \vec{V}_{NP}, \vec{P}_{NP})\}$$

	$\vec{X}$	$\vec{V}$	$\vec{P}$
particle 1$(\vec{X}_1, \vec{V}_1, \vec{P}_1)$	$x_{1,1}, x_{1,2}, \ldots, x_{1,D}$	$v_{1,1}, v_{1,2}, \ldots, v_{1,D}$	$p_{1,1}, p_{1,2}, \ldots, p_{1,D}$
particle 2$(\vec{X}_2, \vec{V}_2, \vec{P}_2)$	$x_{2,1}, x_{2,2}, \ldots, x_{2,D}$	$v_{2,1}, v_{2,2}, \ldots, v_{2,D}$	$p_{2,1}, p_{2,2}, \ldots, p_{2,D}$
	$\ldots$	$\ldots$	$\ldots$
	$\ldots$	$\ldots$	$\ldots$
	$\ldots$	$\ldots$	$\ldots$
particle NP$(\vec{X}_{NP}, \vec{V}_{NP}, \vec{P}_{NP})$	$x_{NP,1}, x_{NP,2}, \ldots, x_{NP,D}$	$v_{NP,1}, v_{NP,2}, \ldots, v_{NP,D}$	$p_{NP,1}, p_{NP,2}, \ldots, p_{NP,D}$

real solution vectors

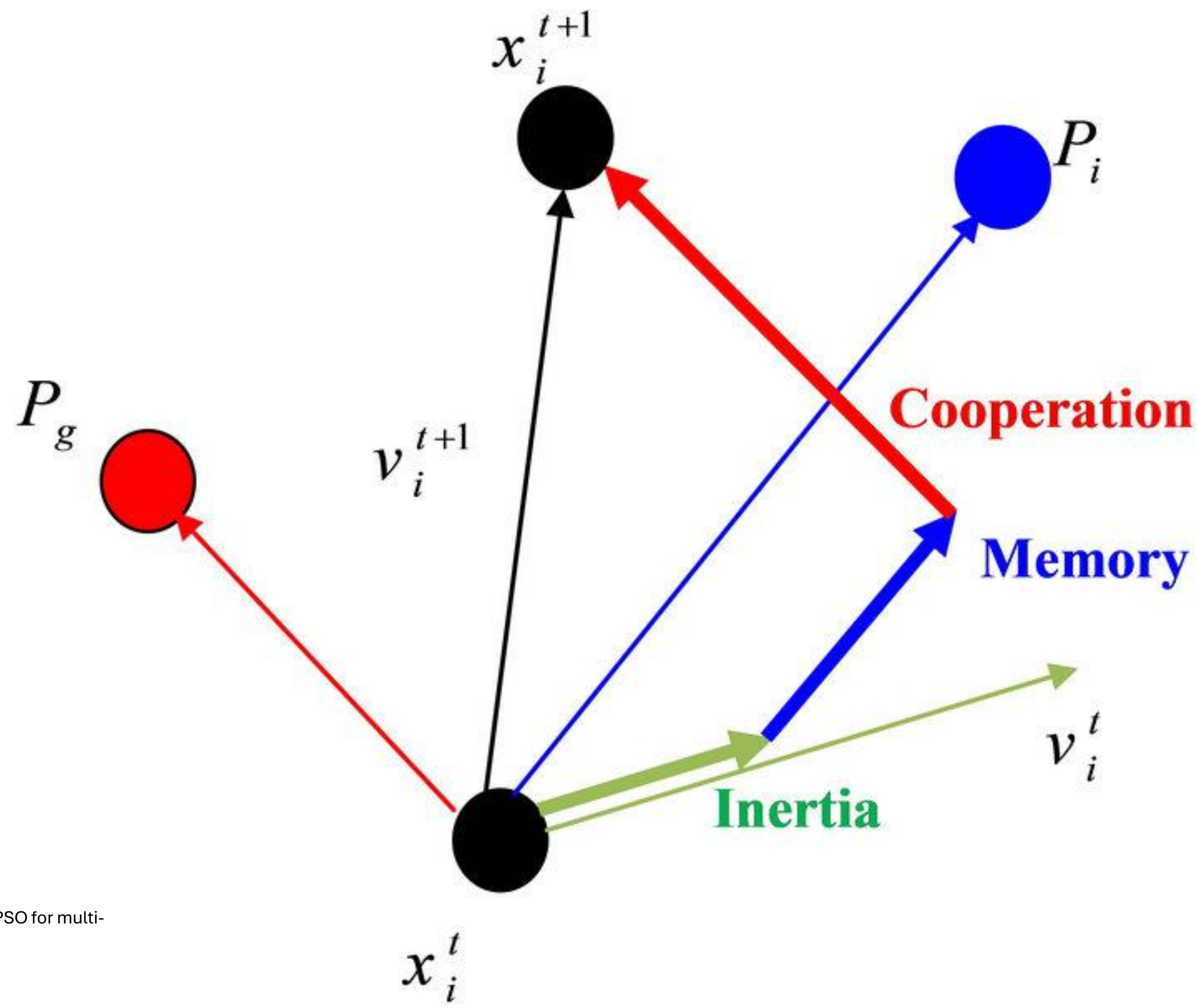
vectors used for the solution update

Particle Movement (solution update)

- The **position** of the **particle $\boldsymbol{i}$** is **updated** by adding its **updated velocity** to its current position. First velocity is updated, then position is changed. Below is the formula for particle i:
  - $\vec{V_i} = w\vec{V_i} + c_1 r_1(\vec{P_i} - \vec{X_i}) + c_2 r_2(\vec{G} - \vec{X_i})$
  - $\overrightarrow{X_i} = \overrightarrow{X_i} + \overrightarrow{V_i}$

- $\vec{G}$ represents the **global best-known position** vector (the best position achieved by the swarm).
- w parameter is the **inertia weight**.
- $c1$ parameter is the **cognitive coefficient**.
- $c2$ parameter is the **social coefficient**.
- $r1$ and $r2$ are uniform random numbers between $[0,1]$.

Particle Movement Visual Interpretation (solution update)



Source: El-Sawy, A. A., Hendawy, Z. M., & El-Shorbagy, M. A. (2013). Reference point based TR-PSO for multi-objective environmental/economic dispatch.

Basic PSO Algorithm

for each particle i = 1, ..., NP **do**

  Initialize the particle's position with a uniformly distributed random vector: ($\vec{X_i}$ ~ U($l_{i,}u_i$)

  Initialize the particle's best known position to its initial position: $\vec{P_i} \leftarrow \vec{X_i}$

  if f($\vec{P_i}$) < f($\vec{G}$) **then**

    update the swarm's best known position: $\vec{G} \leftarrow \vec{P_i}$

  Initialize the particle's velocity: $\vec{V_i}$ ~ U(-| u_i - l_i |, | u_i - l_i |)

while a termination criterion is not met **do**:

  for each particle i = 1, ..., NP **do**

    for each dimension d = 1, ..., D **do**

      Pick random numbers: r_1, r_2 ~ U(0,1)

      Update the particle's velocity: $\vec{V}_{i,d} = w\vec{V}_{i,d} + c_1 r_1(\vec{P}_{i,d} - \vec{X}_{i,d})$+ $c_2 r_2(\vec{G}_d - \vec{X}_{i,d})$

    Update the particle's position: $\vec{X_i} = \vec{X_i} + \vec{V_i}$

    Correct boundaries (upper and lower bounds) of $\vec{X_i}$

    if f($\vec{X_i}$) < f($\vec{P_i}$) **then**

      Update the particle's best known position: $\vec{P_i} \leftarrow \vec{X_i}$

      if f($\vec{P_i}$) < f($\vec{G}$) then

        Update the swarm's best known position: $\vec{G} \leftarrow \vec{P_i}$

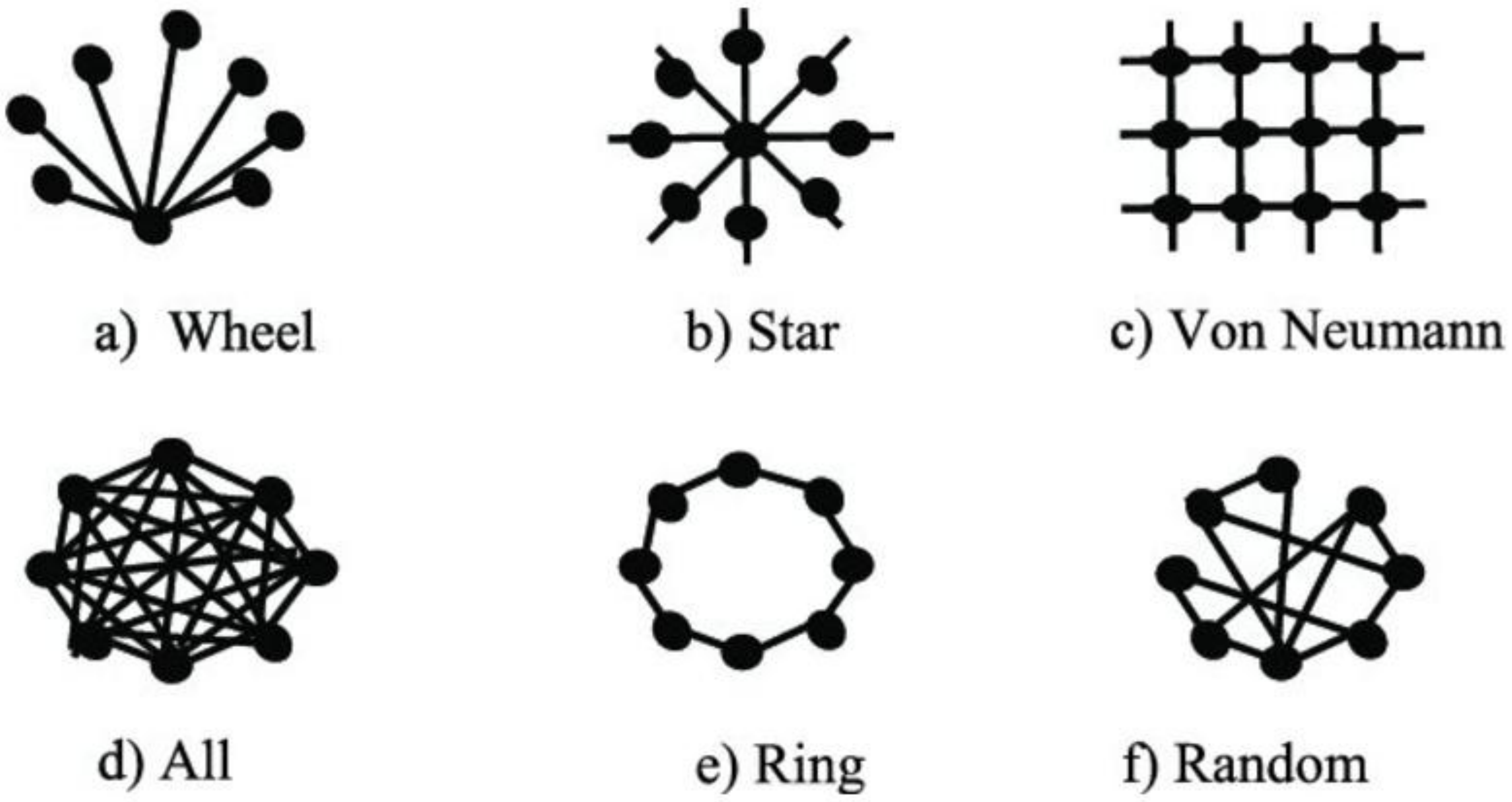
Parameter selection

- $w \in [0,1]$ ensures that the particles' velocities decrease over time, allowing them to converge towards optimal solutions rather than moving uncontrollably away from the search space.
- It is typical to use $c_1, c_2 \in [1,3]$.
- Parameters need to be tuned for a given optimization problem.

Neighbourhoods and topologies

- The swarm's **topology** determines which particles can **exchange information** with one another.
- In the **standard** version of the algorithm, a **global topology** is used as the swarm's communication structure.
- The global topology enables all particles to communicate freely, allowing the entire swarm to share the best position **G** identified by any single particle.
- However, this unrestricted communication may cause the swarm to become trapped in a local minimum.
- To address this, alternative topologies have been explored to regulate the flow of information among particles.

PSO Topologies



Source: Kassoul, K., Zufferey, N., Cheikhrouhou, N., & Belhaouari, S. B. (2022). Exponential particle swarm optimization for global optimization. *IEEE access*, *10*, 78320-78344.