Lecture 5
Swarm Intelligence

Ning Xiong

Mälardalen University

Swarm Intelligence (SI)

- Swarm intelligence (SI) simulates biological systems, as ant colonies or birds flocking.

- SI is composed of simple agents (also called boids) that will interact locally between each other.

Family of Swarm Intelligence

- **<u>Particle Swarm Optimization (PSO)</u>**
- **<u>Ant Colony Optimization (ACO)</u>**
- Artificial Bee Colony (ABC)
- Firefly Algorithm (FA)
- Cuckoo Search (CS)
- ...

PSO idea

Inspired from the nature of social behavior and dynamic movements with communications of insects, birds and fish

PSO: concept

- Uses a number of agents (**particles**) that constitute a swarm moving around in the search space looking for the best solution

- Each particle in search space adjusts its "flying" according to its own flying experience as well as the flying experience of other particles

PSO concept

- A collection of flying particles
- A position of particle corresponds to a possible solution
- The particles move to the global optimal solution
- PSO is used to solve continuous optimization problems

How particles behave in PSO

- Each particle keeps tracks of:
  - its best solution, personal best, *pbest*
  - the best solution of other particles, global best, *gbest*
- Each particle adjusts its position according to
  - its current position
  - its current velocity
  - the difference between its current position and *pbest*
  - the difference between its current position and *gbest*

Particle update rule

- **Particle update rule**

$$p = p + v$$

  - **with**

$$v = v + c_1 * rand * (pBest - p) + c_2 * rand * (gBest - p)$$

cognitive social

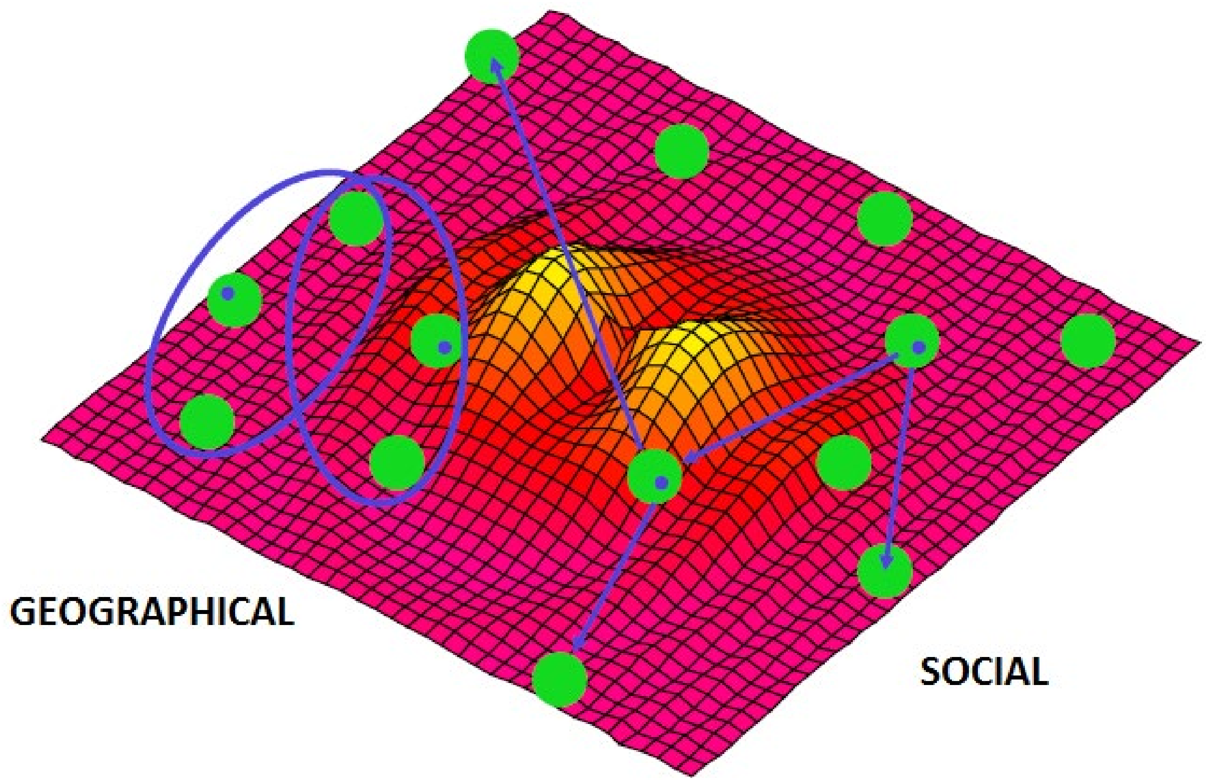
- *p*: particle's position
- *v*: particle's velocity
- c_1: weight of cognitive behavior
- c_2: weight of social behavior
- *pBest*: best position of the particle
- *gBest*: best position of the swarm (sometimes it can be neigbors)
- *rand*: random variable between 0 and 1

Topology of the particle population

- The topologies define the neighbor of every individual. The particle will always belong to its neighbor.

- There are two types of topologies:
  - **Geographical**: The distances between the current particle and the rest are calculated and the neighbors of the particle will be composed of the closer ones.
  - **Social:** The neighbors of a particle are predefined, no matter the position.

Topology of the particle population



Initialization of the particles

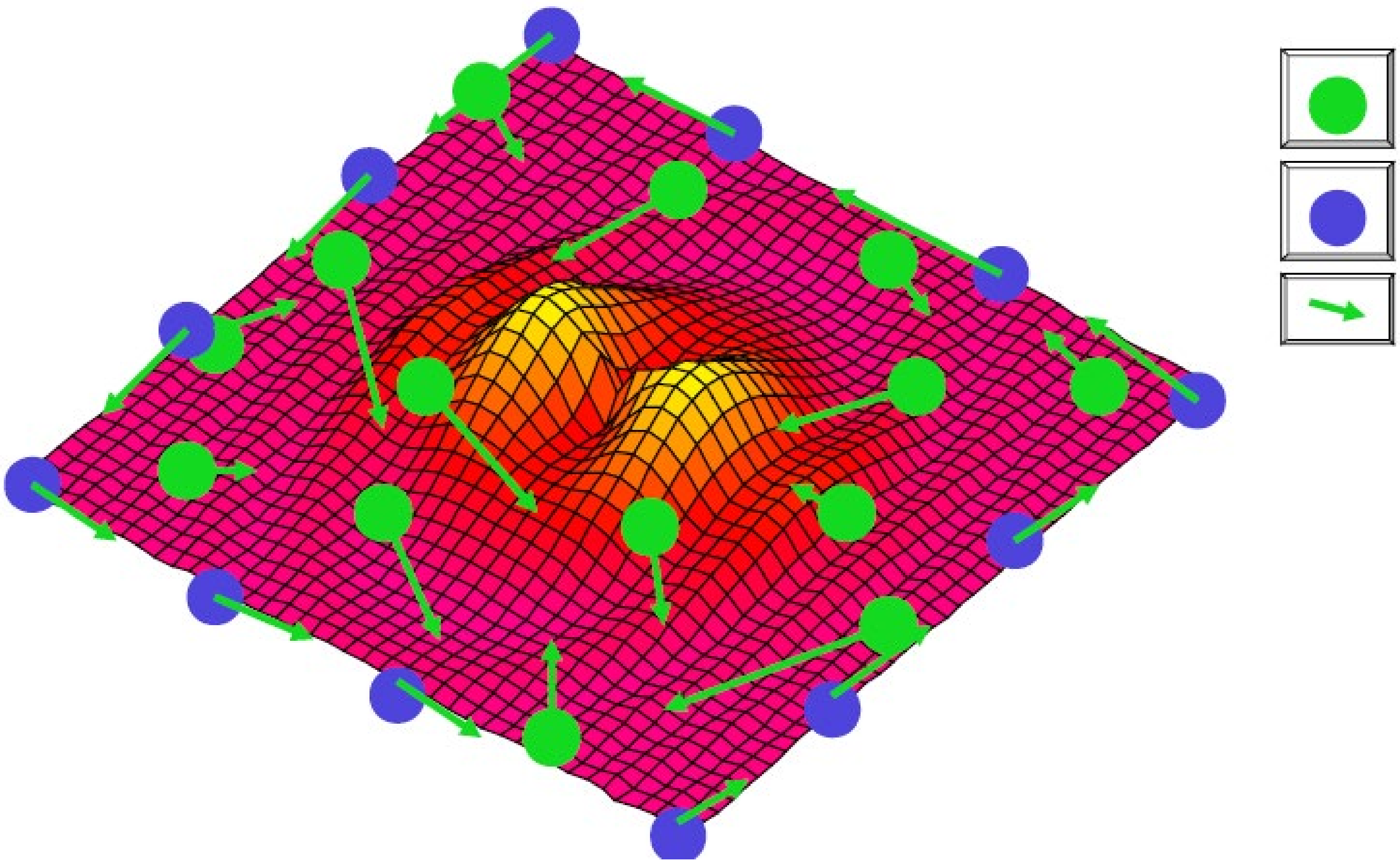
- The positions of the particles are generated randomly in the search space.

- The velocities are randomly generated, with every component in the interval [-Vmax, Vmax]

Do not initializate them to 0, it does not give good results.

- All particles are evaluated to identify pbest and gbest

Initialization of the particles



Movement of the particles

- The Velocity (V) will be added to the position (X), to get the new vector position:

$$X_i \leftarrow X_i + V_i$$

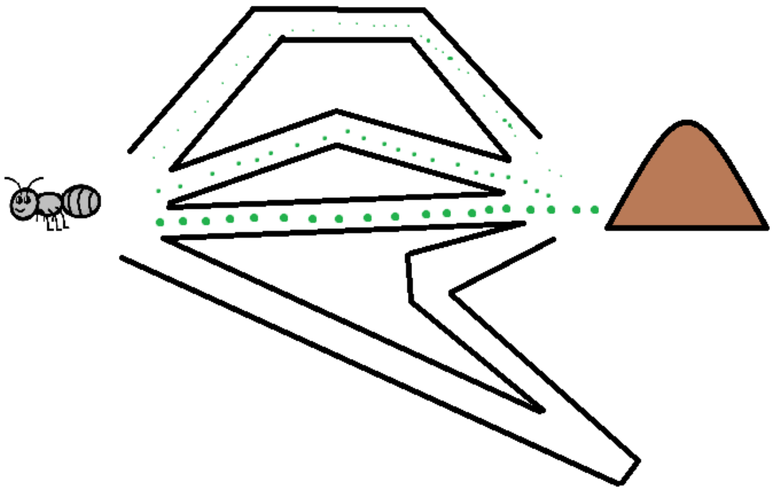
- After getting the new position of the particle, we need to evaluate it. If the new fitness is better than the fitness of pbest, pBest_fitness, then do the following updates

$$pBest_i \leftarrow X_i \quad ; \quad pBest\_fitness \leftarrow x\_fitness.$$

Ant Colony Optimization (ACO)

- ACO is inspired from the behavior of real ants

- Ants can find the shortest path between their nest and a food source using pheromone trails



- ACO is a powerful method to solve combinatorial optimization and graph search problems

Real ants (biological inspiration)

- Real ants find the shortest path between food and nest
- They are almost blind
- They lay pheromone trials on the ground, as signs for other ants (STIGMERGY)
- An ant decides which pheromone trail to follow

  with probability but based on pheromone strength

- An ant lay pheromone on the trail it follows, thus reinforcing the trail
- The more ants follow a trail, the more pheromone it receives, and higher chance ants will follow it
- Pheromone strength on trails decay with time

Stigmergy in Humans

ACO algorithm

- The ACO algorithm imitates the real ants in an artificial ant colony to find the minimum path on the graph

- Each artificial ant has a probabilistic mechanism to build its solution by using

  - Pheromone, **τ** (tau), that change through the time to reflect the experience of the agents.

  - Heuristic information, **η**, about the problem being solved.

Artificial Ant: Probabilistic rule

- The most common probabilistic rule defines the probability in which the ant will move from node **r** to node **s** as:

$$P(r,s) = \begin{cases} \dfrac{[\tau_{rs}]^{\alpha} * [\eta_{rs}]^{\beta}}{\sum_{u \in J_k(r)} [\tau_{ru}]^{\alpha} * [\eta_{ru}]^{\beta}} & if\ s \in J_k(r) \\ 0, & in\ other\ case \end{cases}$$

where:

- τ_{rs} is the pheromone of the edge e_{rs}
- η_{rs} is the heuristic information guiding the path construction.

  Greedy case: $\eta_{rs} = \frac{1}{d_{rs}}$ meaning nearest node most desirable
- $J_k(r)$ is the set of nodes connected to **r** but not visited by the ant.
- α and β are weights representing the trade off of the global and local factors of finding the path

Update the pheromone

- After all ants complete the paths, pheromone will be updated as:

$$\tau_{rs}(t) = (1 - p) * \tau_{rs}(t - 1) + \Delta\tau_{rs}$$

- Pheromone evaporates a small amount after each iteration, with $0<p<1$ being an evaporation constant

- The density of pheromone laid on the edge (r, s) by the *m* ants is

$$\Delta\tau_{rs} = \sum_{k=1}^{m} \Delta\tau_{rs}^{k}$$

where:

$$\Delta\tau_{rs}^{k} = \begin{cases} \dfrac{1}{C(S_k)}, \text{if the ant } k \text{ visited the edge} \\ 0, \quad \text{otherwise} \end{cases}$$

$C(S_k)$ is the path length generated by the ant **k**.

ACO algorithm description (TSP)

- Initialize the pheromone intensity on the edges
- Position ants at different cities
- Each ant moves from city to city based on probability
- After all ants complete their paths, calculate their length and save the shortest path. Update the pheromone on all edges
- Iterate untill the maximum number of iterations or covergence

Help for the TSP problem (lab)

- You have a distance D matrix:

	1	2	3	4	5	6
1	∞	1	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$
2	1	∞	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$
3	$\sqrt{5}$	$\sqrt{2}$	∞	$\sqrt{2}$	$\sqrt{5}$	3
4	$\sqrt{5}$	2	$\sqrt{2}$	∞	1	$\sqrt{5}$
5	2	$\sqrt{5}$	$\sqrt{5}$	1	∞	$\sqrt{2}$
6	$\sqrt{2}$	$\sqrt{5}$	3	$\sqrt{5}$	$\sqrt{2}$	∞

Help for the TSP problem (lab)

- The η matrix should be calculated as follows:

	1	2	3	4	5	6
1	-	1	0.447	0.447	0.5	0.707
2	1	-	0.707	0.5	0.447	0.447
3	0.447	0.707	-	0.707	0.447	0.333
4	0.447	0.5	0.707	-	1	0.447
5	0.5	0.447	0.447	1	-	0.707
6	0.707	0.447	0.333	0.447	0.707	-

Help for the TSP problem (lab)

- The initial pheromone table is suggested like this (all τ values are initialized to 10 for example):

	1	2	3	4	5	6
1	-	10	10	10	10	10
2	10	-	10	10	10	10
3	10	10	-	10	10	10
4	10	10	10	-	10	10
5	10	10	10	10	-	10
6	10	10	10	10	10	-

Extra reading

- Particle Swarm Optimization
  - Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *evolutionary computation, 2001. Proceedings of the 2001 Congress on* (Vol. 1, pp. 81-86). IEEE.

- Ant Colony Optimization
  - M. Dorigo, V. Maniezzo, A. Colorni. Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and cybernetics, Part B, Vol 26, 29-41