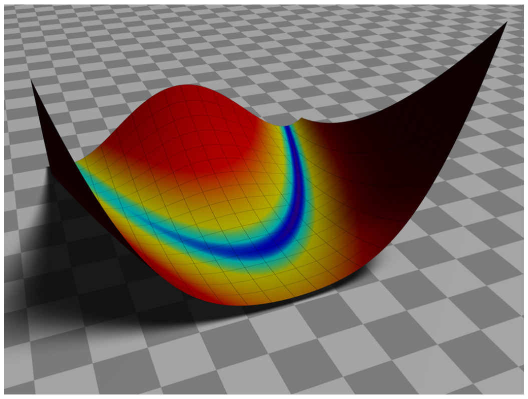
Particle Swarm

Lecture 26



ME EN 575
Andrew Ning
aning@byu.edu

Outline

Discuss with a partner some ideas for derivative-free optimization (could be bio-inspired or not).

Particle Swarm

Basic Concept

Think of a swarm of insects (e.g., bees).

Each agent has the following three characteristics:

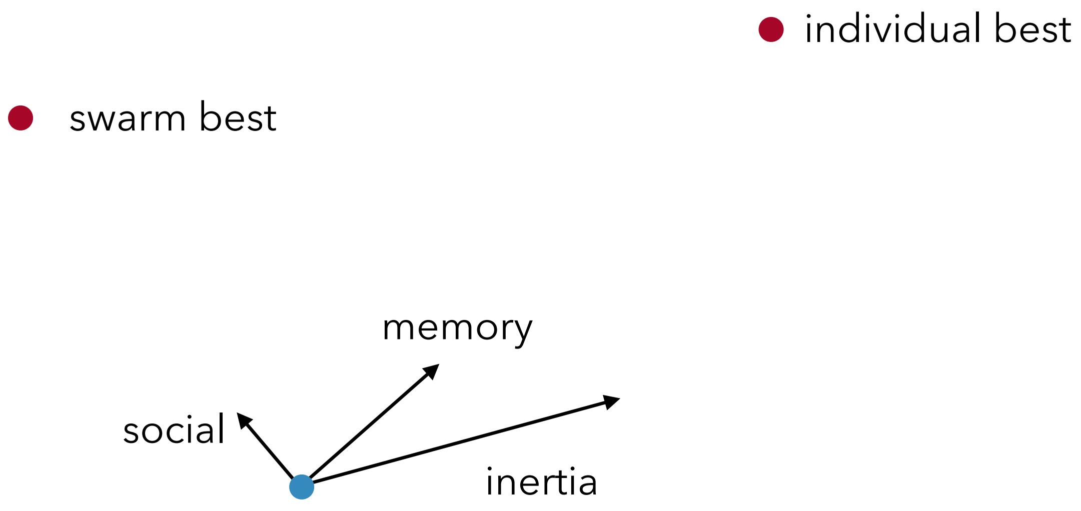
- Momentum. Agent will tend to continue in direction it is headed.
- Memory. Agent will remember the best place it has visited, and will want to return towards that position.
- Social Influence. Agent will want to visit the best place the swarm (or neighbors) has found.

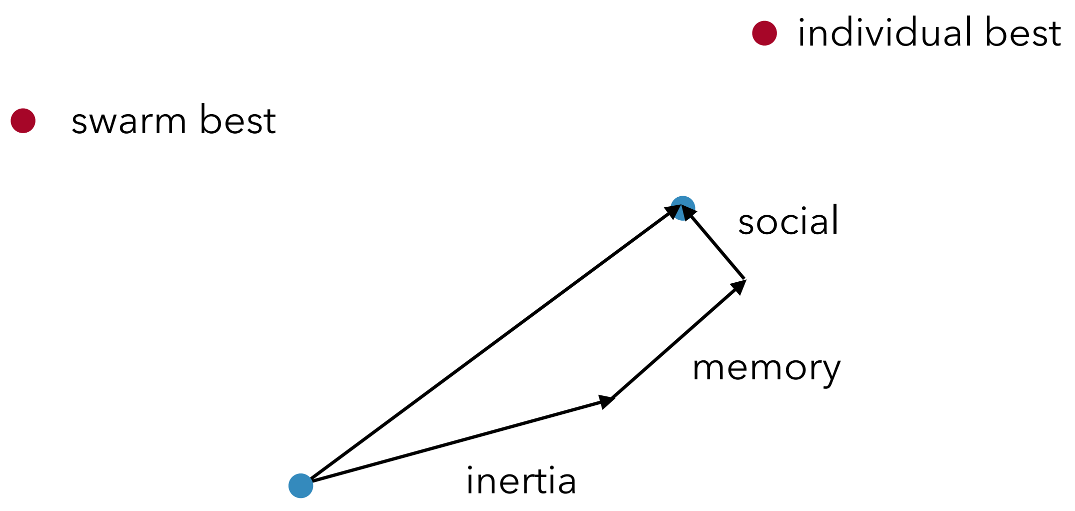
View Animation

Overall Algorithm

1. Initialize positions *and velocities* of the population
2. Repeat until converged
   2.1 Evaluate fitness value
   2.2 Update best agent position and best swarm position
   2.3 Update position of each particle (and enforce bounds)

Velocity Update





Inertia

$$v_{k+1_i} = w v_k$$

$w \in [0, 1.2]$ is an inertial parameter (typically 0.8–1.2).

How would changing w affect the optimization behavior?

Memory

Best individual point: x_k^i

A "velocity" towards best individual point: $(x_k^i - x_k)/\Delta t$

$$v_{k+1_m} = c_1 \frac{(x_k^i - x_k)}{\Delta t}$$

c_1 is a random number between 0 and c_{1max} (usually around 2.05). Can be a scalar or a vector, but usually better as a vector.

Social

Best group point (could be a neighborhood or entire swarm): x_k^g

A "velocity" towards best swarm point: $(x_k^g - x_k)/\Delta t$

$$v_{k+1_s} = c_2 \frac{(x_k^g - x_k)}{\Delta t}$$

c_2 is a random number between 0 and c_{2max} (usually around 2.05). Can be a scalar or a vector, but usually better as a vector.

Total velocity

$$\begin{aligned} v_{k+1} &= v_{k+1_i} + v_{k+1_m} + v_{k+1_s} \\ &= w v_k + c_1 \frac{(x_k^i - x_k)}{\Delta t} + c_2 \frac{(x_k^g - x_k)}{\Delta t} \end{aligned}$$

Position update:

$$x_{k+1} = x_k + v_{k+1} \Delta t$$

Time dependence is artificial. Eliminate:

$$\Delta x_{k+1} = w \Delta x_k + c_1 (x_k^i - x_k) + c_2 (x_k^s - x_k)$$

$$x_{k+1} = x_k + \Delta x_{k+1}$$

(remember to enforce bounds)

Other Considerations

Generally, some form of velocity clamping and velocity damping is needed.

Velocity clamping means we put an upper bound on the velocity so that we don't move too far. Δx of around 10–20% of range is reasonable.

Velocity damping means that at each iteration we decrease what that maximum bound is $(e.g., V_{max} = 0.9V_{max})$, or decrease the inertial parameter w, to improve convergence.

Convergence Criteria:

- Best value stops changing
- velocity decreases below some tolerance
- distance between every particle and best is below some tolerance
- distance between best and worst is below some tolerance

Example

Review all three methods with example animations.