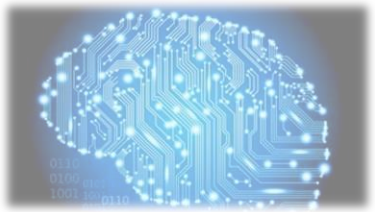
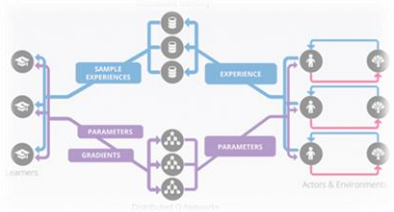



Particle Swarm Optimization

Seolhee Lee

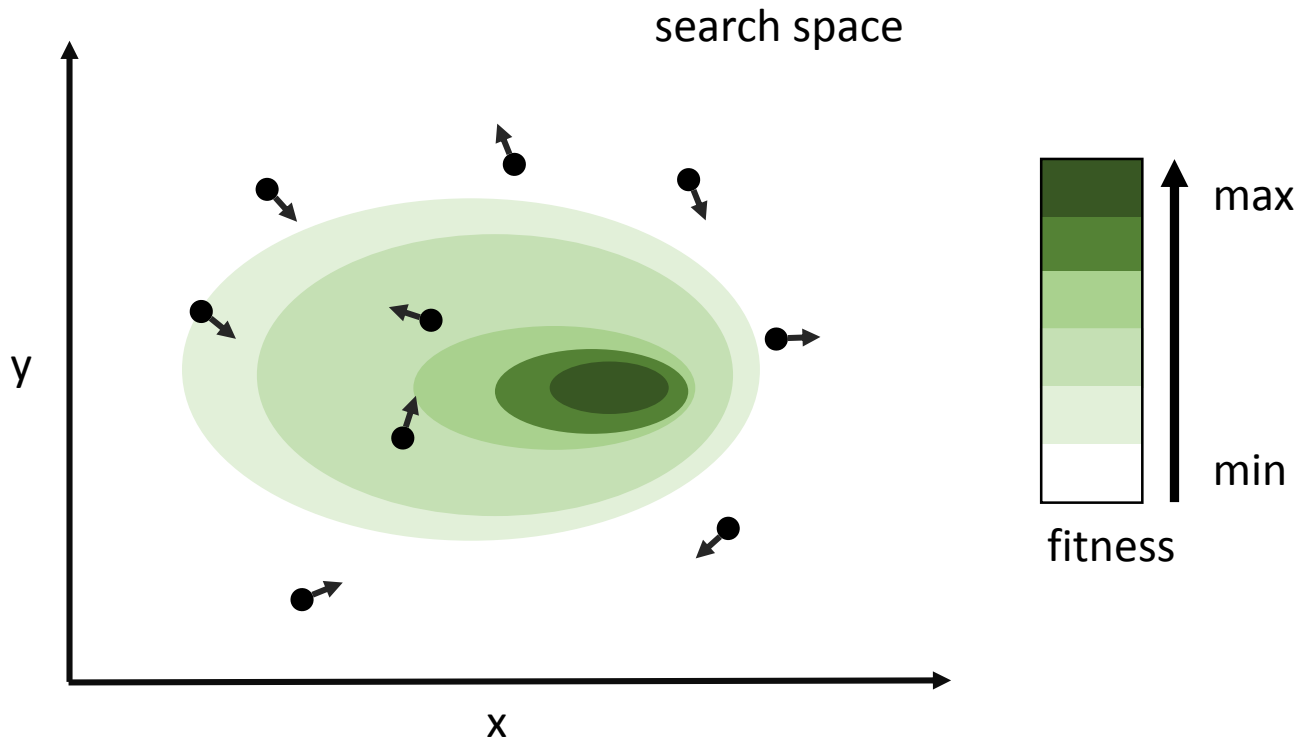
2018. 02. 14.

Introduction

- Particle Swarm Optimization(PSO) is proposed by Kennedy and Eberhart in 1995
- A concept for optimization of nonliner functions using particle swarm methodology
- Inspired by simulation social behavior of bird flocking or fish schooling.

Introduction

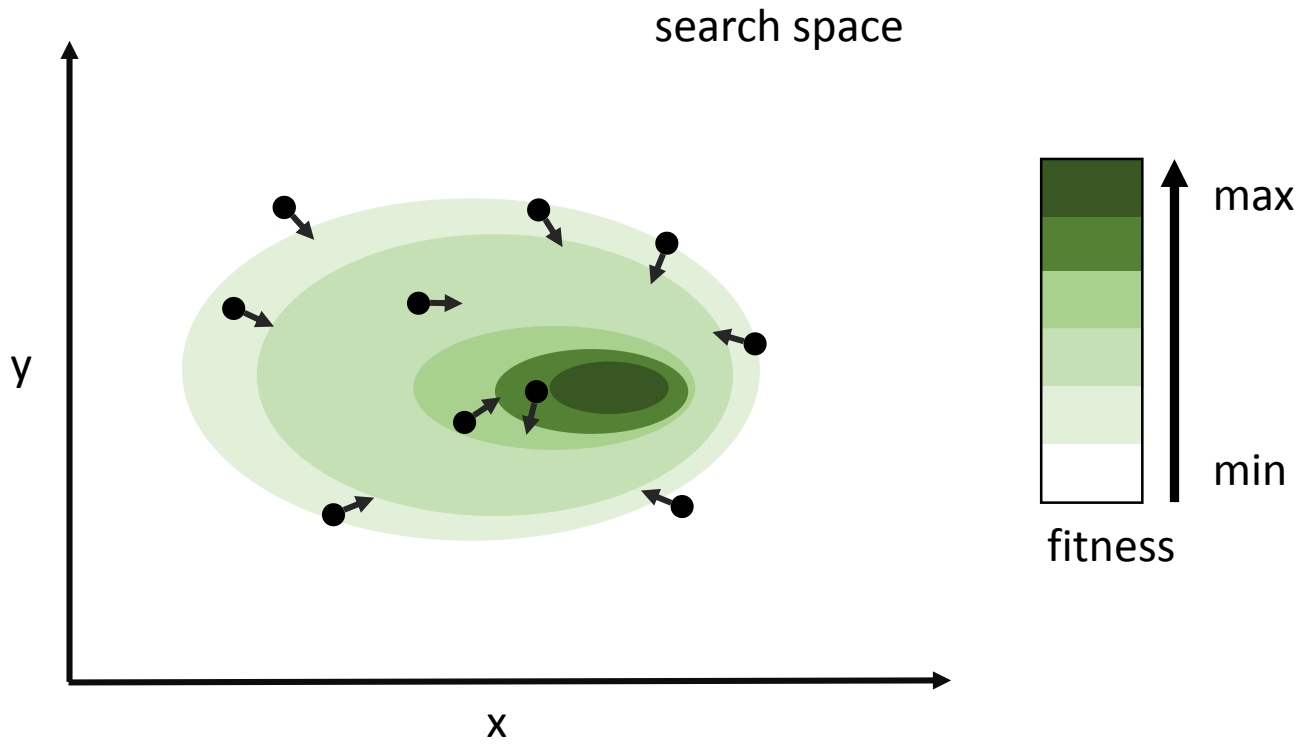
Concept



- Each particle calculates its new velocity based on its own best experience and the best experience of the swarm, and repeats moving particles.

Introduction

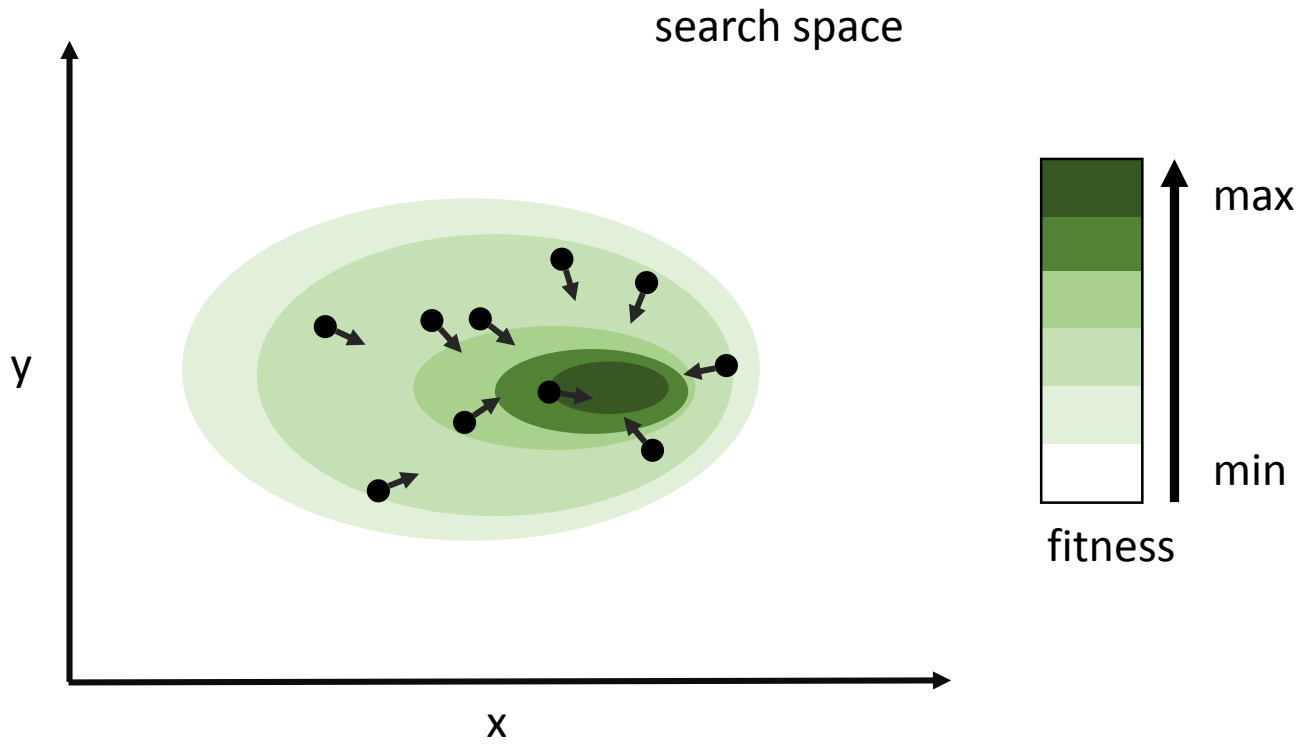
Concept



- Each particle calculates its new velocity based on its own best experience and the best experience of the swarm, and repeats moving particles.

Introduction

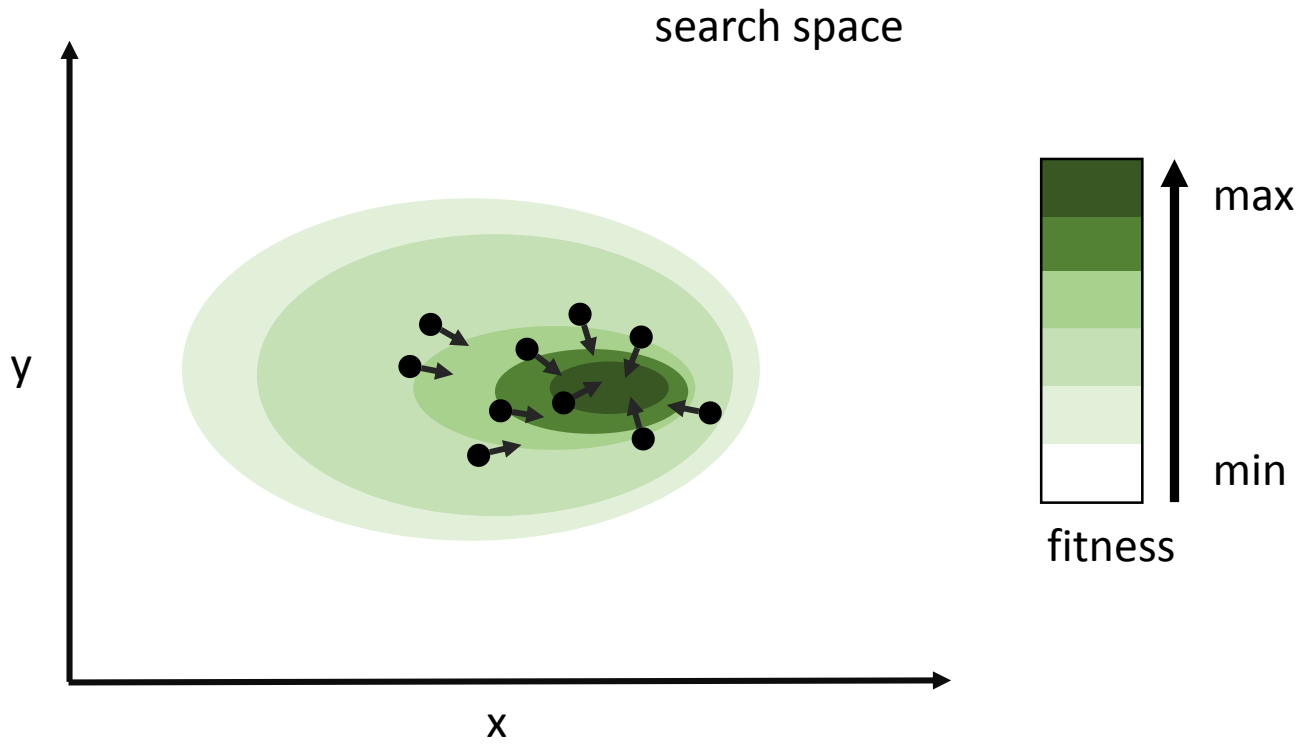
Concept



- Each particle calculates its new velocity based on its own best experience and the best experience of the swarm, and repeats moving particles.

Introduction

Concept



- Each particle calculates its new velocity based on its own best experience and the best experience of the swarm, and repeats moving particles.

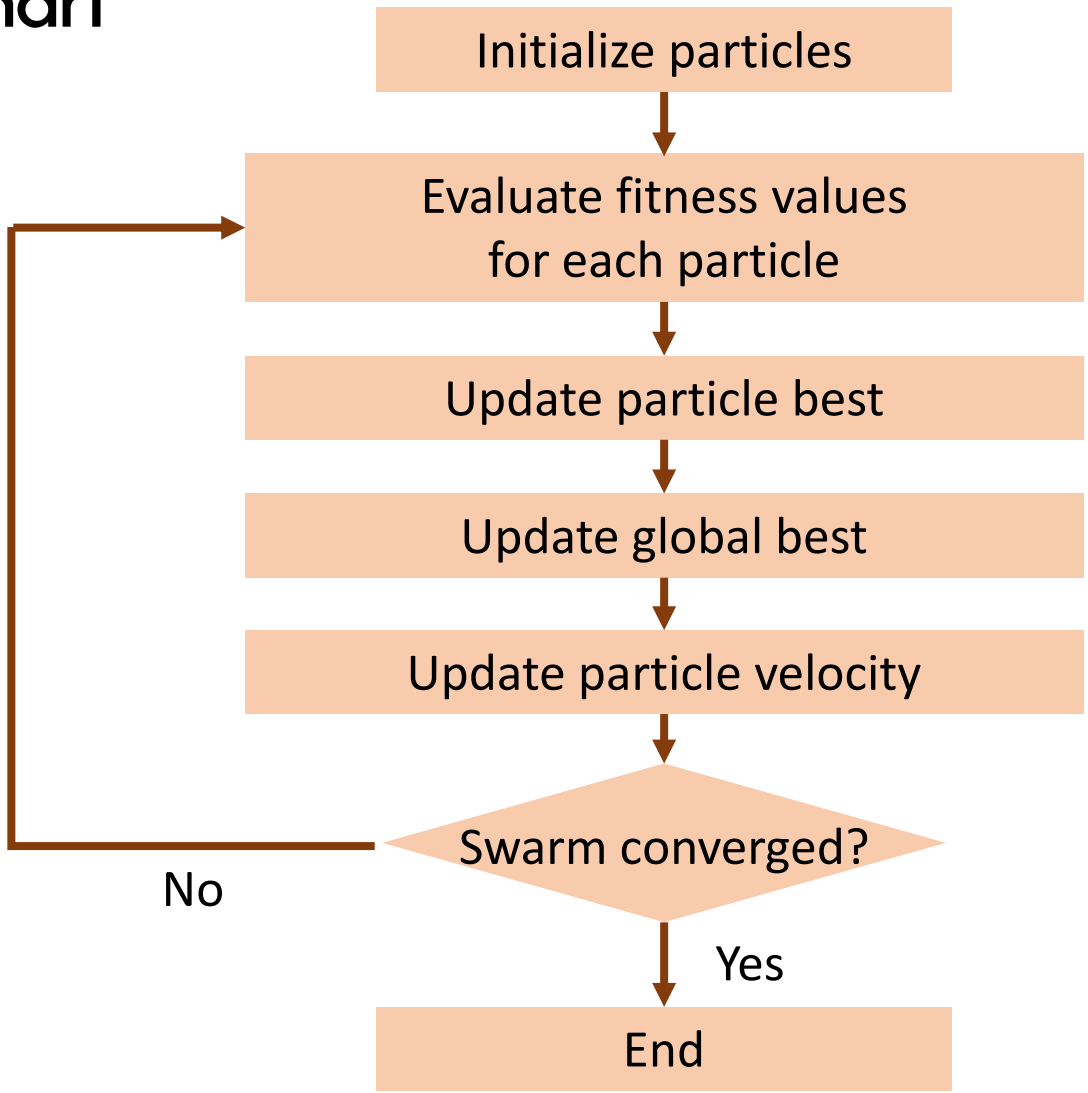
Algorithm

How it works

- PSO is Initialized with random particles
- Particles move through the search space and are evaluated according to the objective function in every iteration.
- Each particle stores personally, globally best value and position (called pbest, gbest)
- Each particle modifies its position according to its current position, velocity, the distance between its current position and pbest, gbest.

Algorithm

Flowchart



Algorithm

Particle update

$$X_i^{n+1} = X_i^n + V_i^{n+1}$$

$$V_i^{n+1} = \underbrace{wV_i^n}_{\text{inertia}} + \underbrace{c_1 r_1^n (P_i^n - X_i^n)}_{\text{Personal influence}} + \underbrace{c_2 r_2^n (G - X_i^n)}_{\text{Social influence}}$$

- X_i: ith particle's position
- V_i: ith particle's velocity
- P_i:best position of ith particle (pbest)
- G:best position of swarm (gbest)
- c_1, c_2: weight related to pbest, gbest
- r_i, r_2: random variables

Algorithm

Particle update

$$X_i^{n+1} = X_i^n + V_i^{n+1}$$

$$V_i^{n+1} = \underbrace{wV_i^n}_{\text{inertia}} + \underbrace{c_1 r_1^n (P_i^n - X_i^n)}_{\text{Personal influence}} + \underbrace{c_2 r_2^n (G - X_i^n)}_{\text{Social influence}}$$

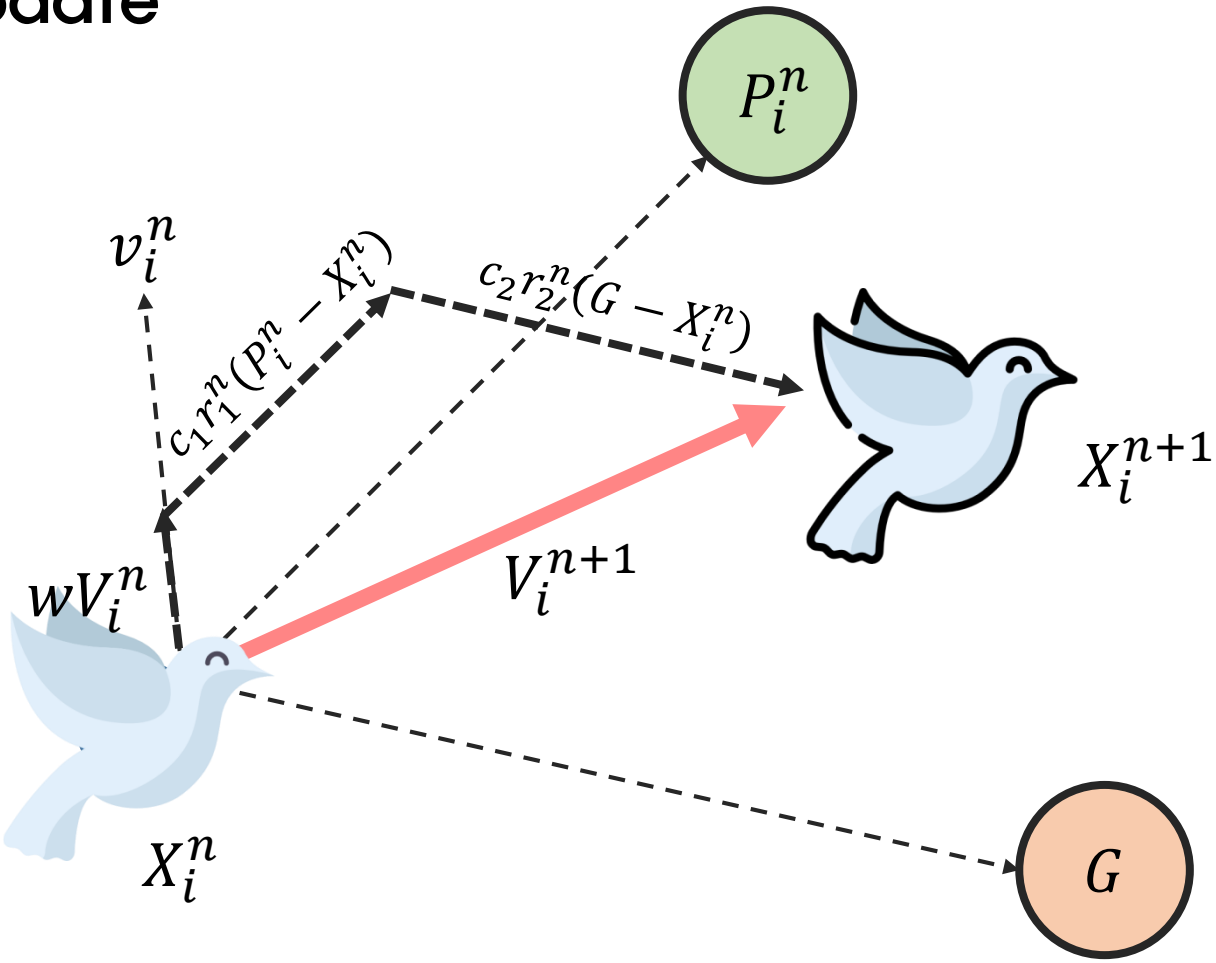
Makes the particle move in the same direction and with the same velocity

Makes the particle return to the place that most satisfied it in the past

Makes the particle follow the best neighbors direction

Algorithm

Particle update



$$V_i^{n+1} = wV_i^n + c_1 r_1^n (P_i^n - X_i^n) + c_2 r_2^n (G - X_i^n)$$

Conclusion

Advantage

- Simple implementation
- Few algorithm parameters
- Global optimization

Disadvantage

- easy to fall into Local minimum