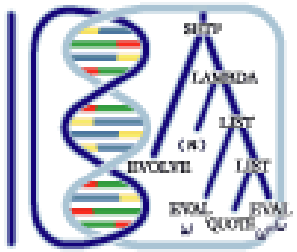
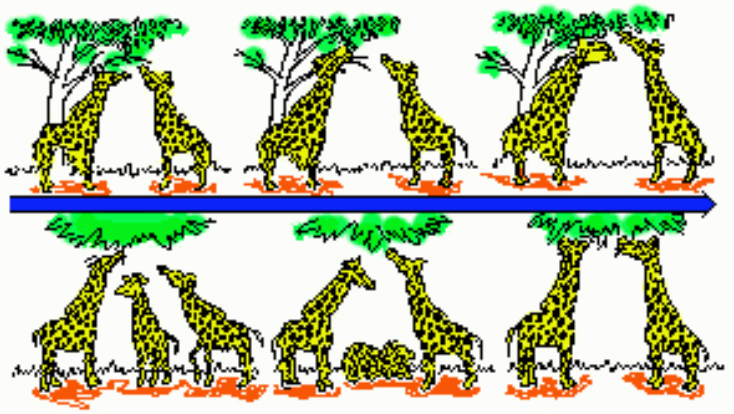


PSO アルゴリズム

東京大学大学院
工学系研究科
電気系工学専攻
伊庭斉志

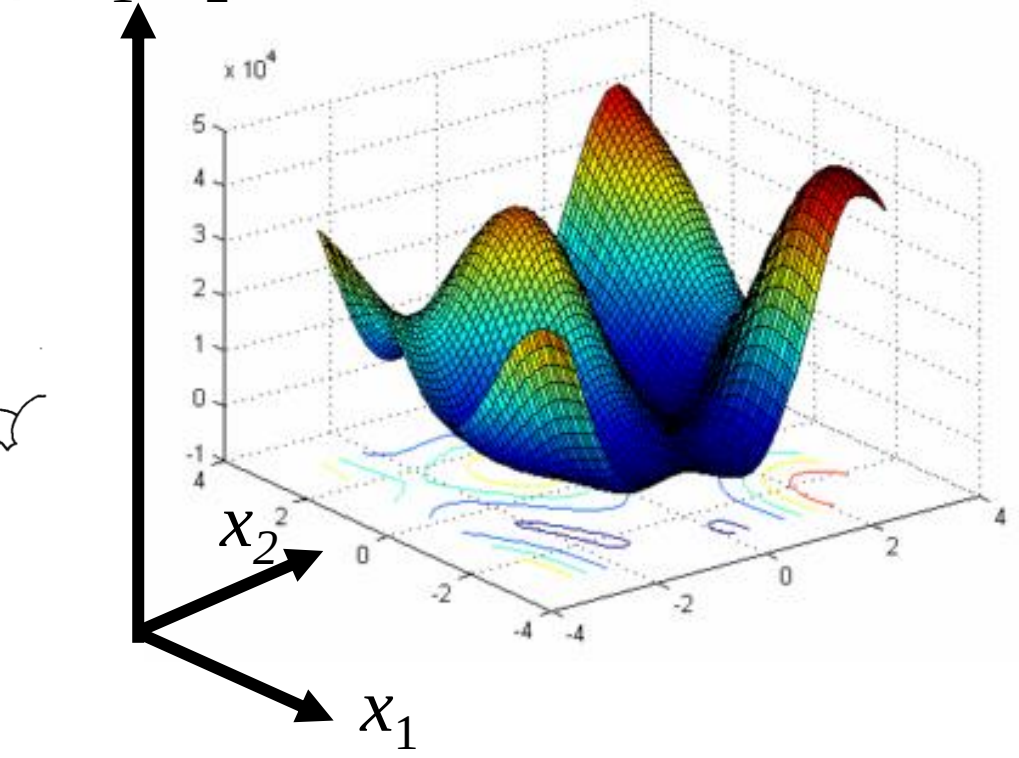


iba lab.

最適値問題を解く

- 最大値を求める問題　　$x \in X$ 内で $\max_x \{f(x)\}$ を与える x を求めよ。
- 工学やAIの最も基本的探索問題

適合度ランドスケープ



Particle Swarm Optimization

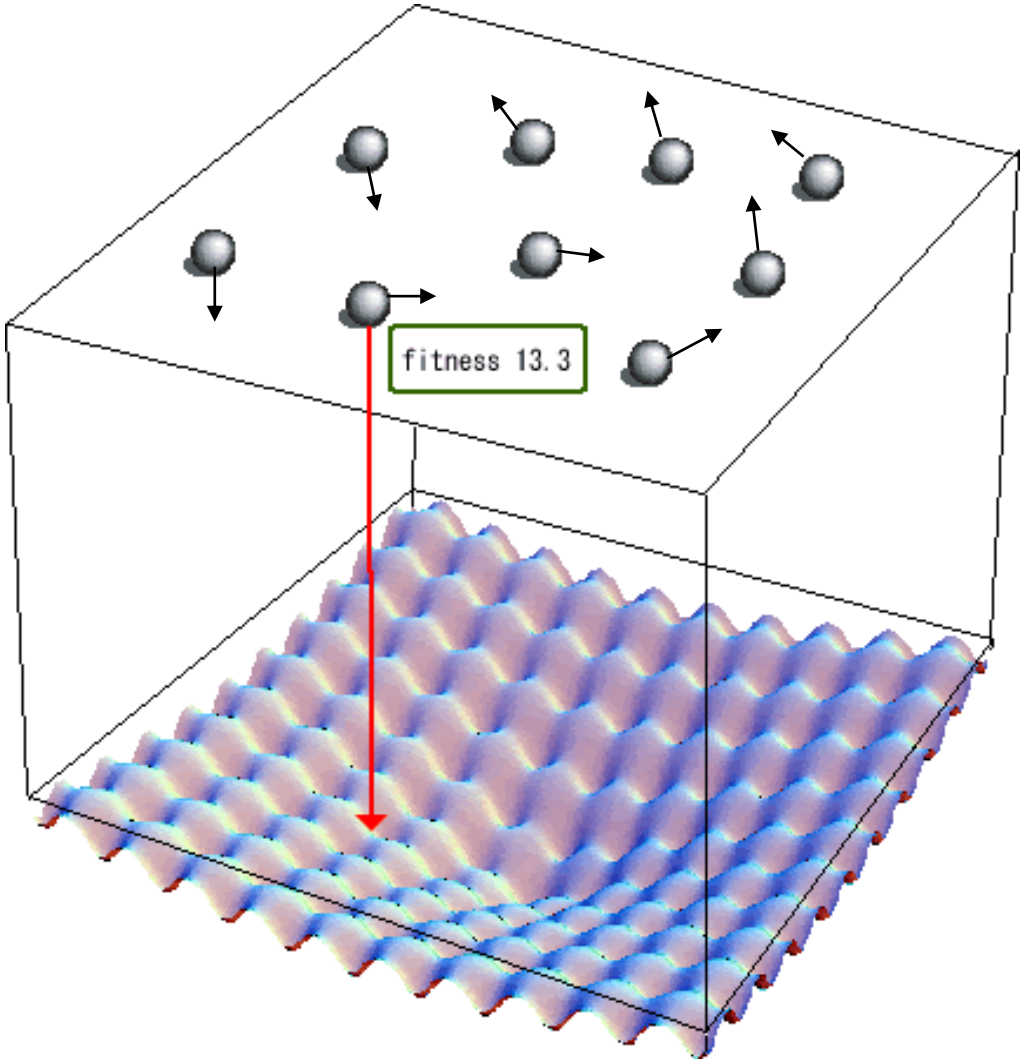
- James Kennedy and Russel C. Eberhart(1995)

```
begin
        initialize
        while(not terminate-condition)do
            begin
                    evaluate
                    calcurale new velocity vectors
                    move
            end
end
```

$$\vec{v_i} = \chi(\omega\vec{v_i} + \psi_1(\vec{p_i} - \vec{x_i}) + \psi_2(\vec{p_g} - \vec{x_i}))$$

$$\vec{x_i} = \vec{x_i} + \vec{v_i}$$

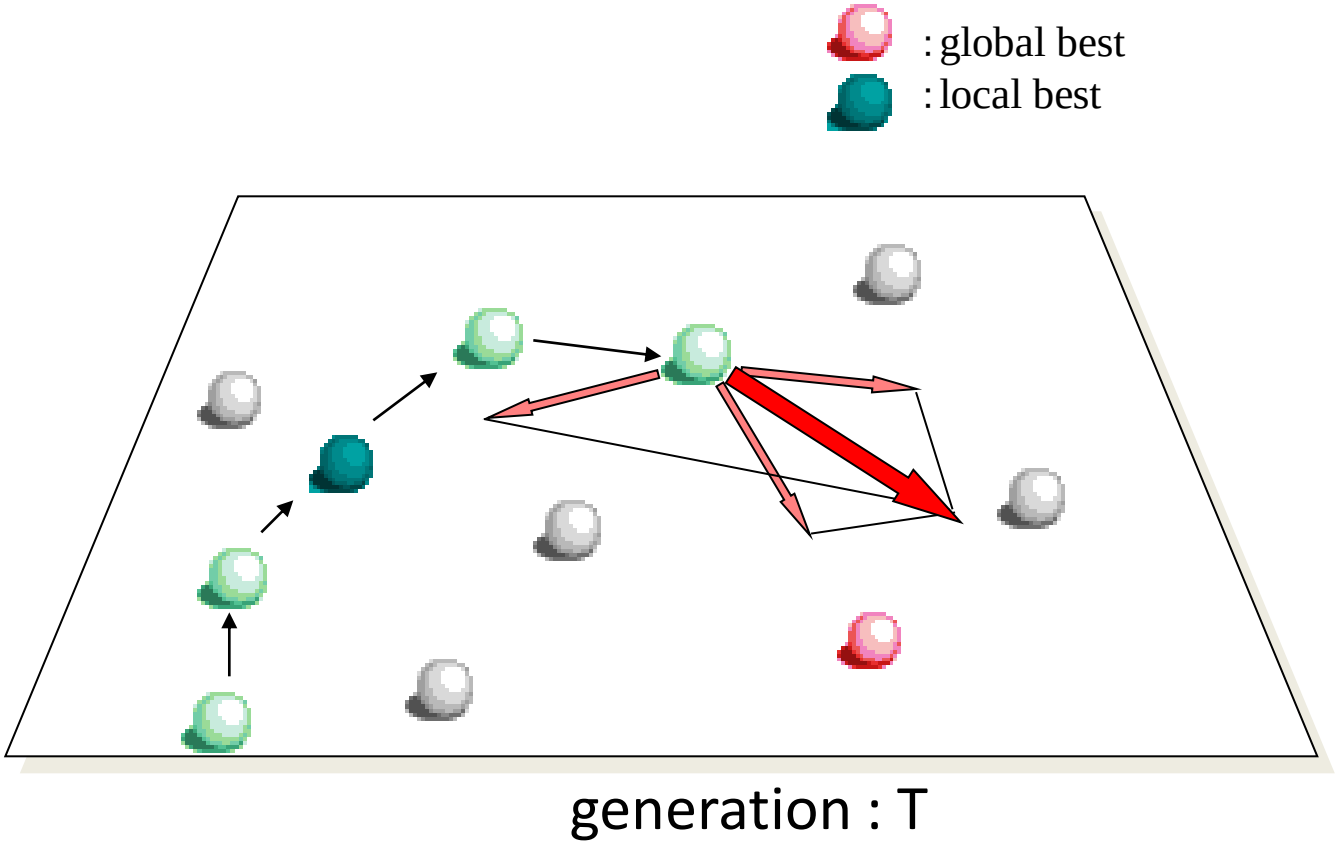
PSO Algorithm



- PSO consists of a number of particles moving around in the search space, each representing a possible solution to numerical problems

PSO Algorithm

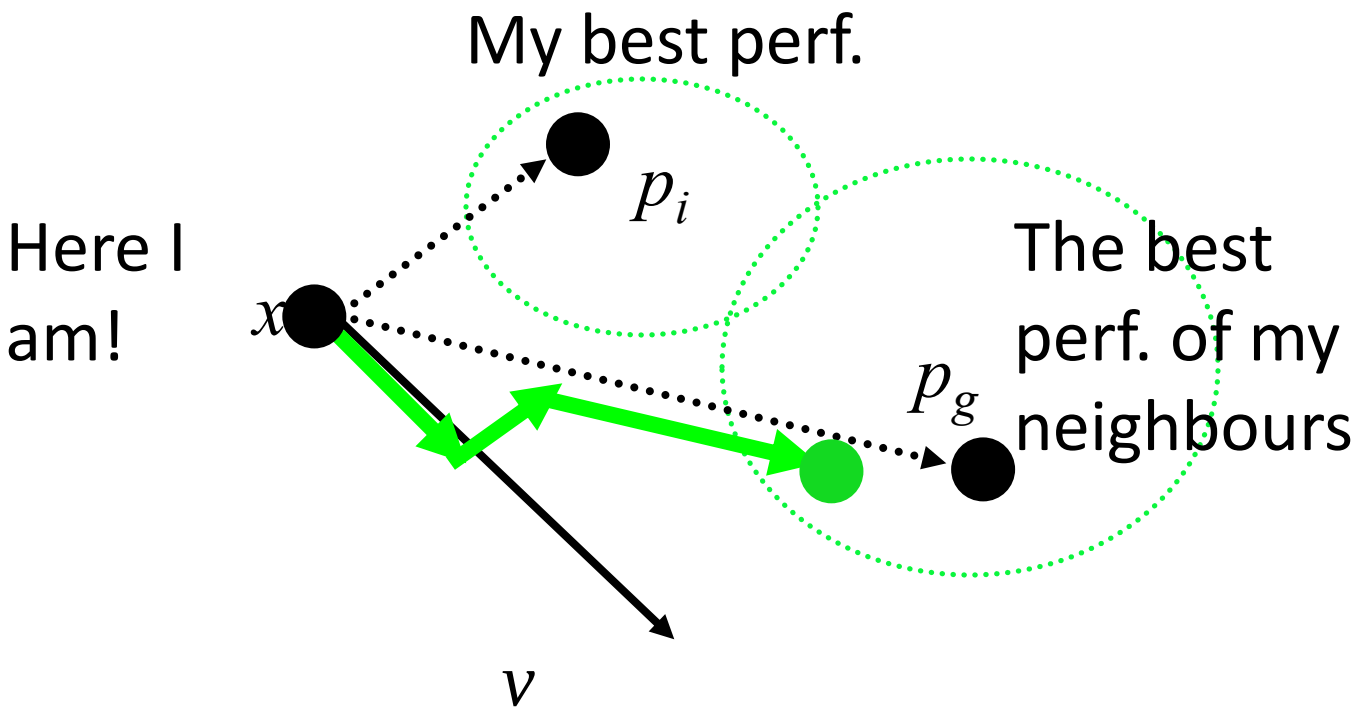
- Particulars about motion of PSO



Particle Swarm Optimization

- Swarm : a set of particles (S)
- Particle: a potential solution
  - Position, $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathbb{R}^n$
  - Velocity , $V_i = (v_{i1}, v_{i2}, \ldots, v_{in}) \in \mathbb{R}^n$
- Each particle maintains
  - Individual best position: $P_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \in \mathbb{R}^n$, $pbest_i = f(P_i)$
- Swarm maintains its global best: $P_g \in \mathbb{R}^n$, $gbest = f(P_g)$

pbest and gbest



PSO Algorithm

- Basic algorithm of PSO:
  1. Initialize the swarm from the solution space
  2. Evaluate fitness of each particle
  3. Update individual and global bests
  4. Update velocity and position of each particle
  5. Go to step 2, and repeat until termination condition

PSO Algorithm (cont.)

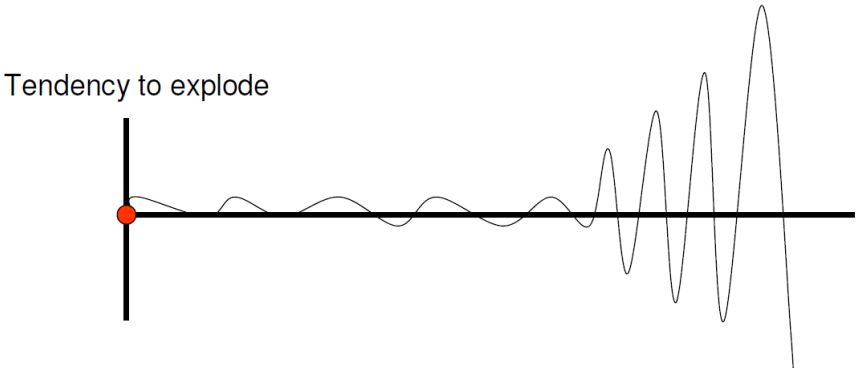
- **Original velocity update equation:**

$$V_i^{t+1} = \underbrace{V_i^t}_{\text{Inertia}} + \underbrace{\varphi_1 . r_1 (P_i - X_i^t)}_{\text{Cognitive Component}} + \underbrace{\varphi_2 . r_2 (P_g - X_i^t)}_{\text{Social Component}}$$

  - with $r_1, r_2 \sim U(0,1)$
  - φ_1, φ_2 : acceleration constant

PSO Algorithm - Parameters

- Acceleration constan φ_1, φ_2
  - Small values limit the movement of the particles
  - Large values : tendency to explode toward infinity
  - In general $\varphi_1 + \varphi_2 \leq 4$



- Maximum velocity
  - Velocity is a stochastic variable => uncontrolled trajectory

$$\text{If} \quad v_{\text{id}} > v_{\text{max}} \text{ then } v_{\text{id}} = v_{\text{max}}$$

$$\text{else if } v_{\text{id}} < -v_{\text{max}} \text{ then } v_{\text{id}} = -v_{\text{max}}$$

Pseudocode

```
Equation (a)
v[] = c0 *v[]
     + c1 * rand() * (pbest[] - present[])
     + c2 * rand() * (gbest[] - present[])
   (in the original method, c0=1, but many
     researchers now play with this parameter)

Equation (b)
present[] = present[] + v[]
```

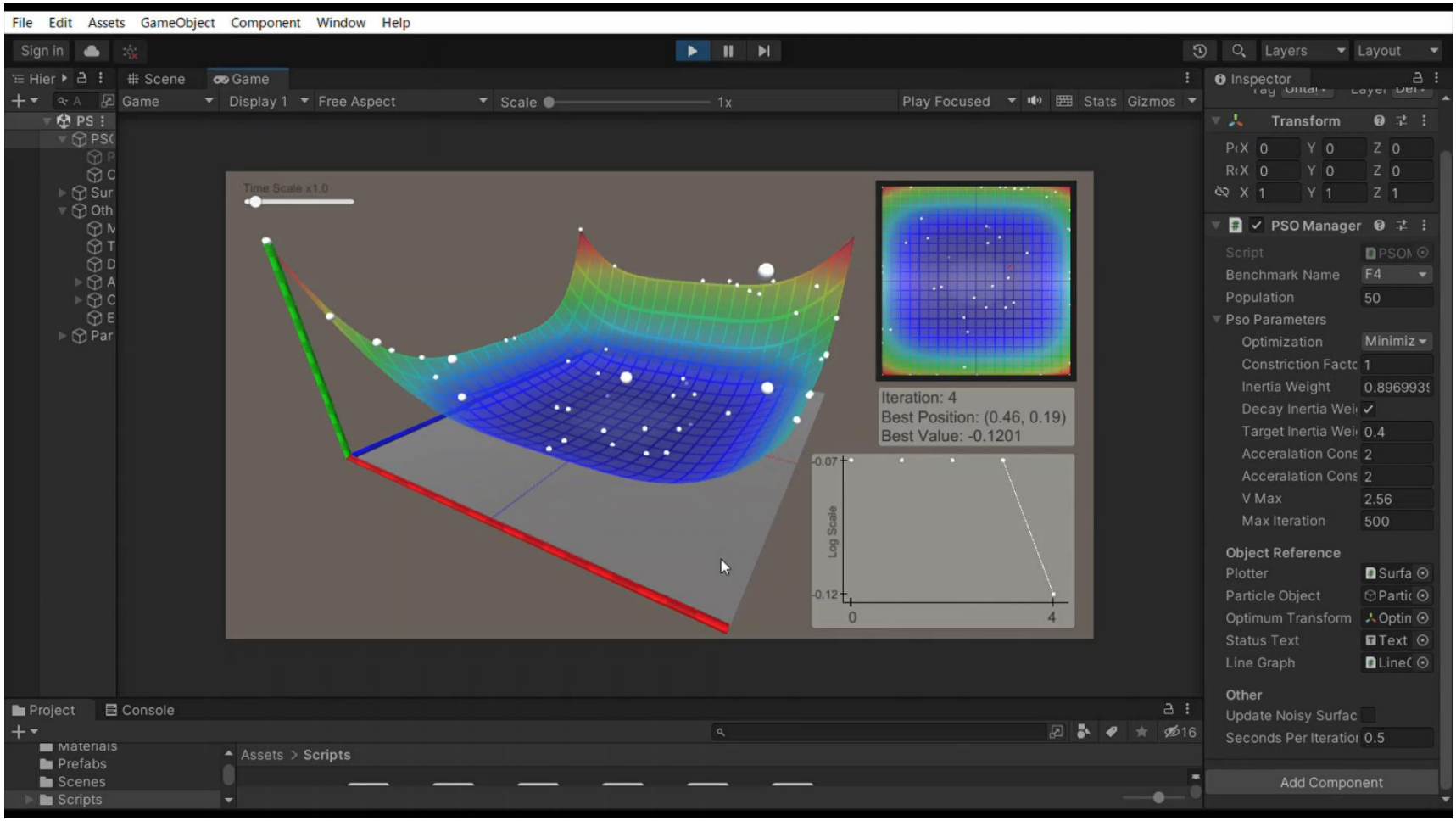
Pseudocode

```
For each particle
    Initialize particle
END

Do
    For each particle
        Calculate fitness value
        If the fitness value is better than its peronal best
        set current value as the new pBest
    End

    Choose the particle with the best fitness value of all as gBest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    End
While maximum iterations or minimum error criteria is not attained
```

PSOのシミュレータ


File Edit Assets GameObject Component Window Help
Sign in
Layers
Layout
Hier
Scene
Game
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Time Scale x1.0
Iteration: 4
Best Position: (0.46, 0.19)
Best Value: -0.1201
Log Scale
-0.07
-0.12
0
4
Inspector
Transform
PSO Manager
Script
Benchmark Name
F4
Population
50
Pso Parameters
Optimization
Minimiz
Constriction Facto
Inertia Weight
Decay Inertia Wei
Target Inertia Wei
0.4
Acceralation Cons
Acceralation Cons
V Max
2.56
Max Iteration
500
Object Reference
Plotter
Particle Object
Optimum Transform
Status Text
Line Graph
Other
Update Noisy Surfac
Seconds Per Iteratio
0.5
Add Component
Project
Console
Assets > Scripts
Prefabs
Scenes
Scripts