

2024 Probabilistic Model Class

# Chapter 5. Population Based Search

## - Particle Swarm Optimization

---



순천향대학교 미래융합기술학과

Senseable AI Lab

석사과정 김병훈

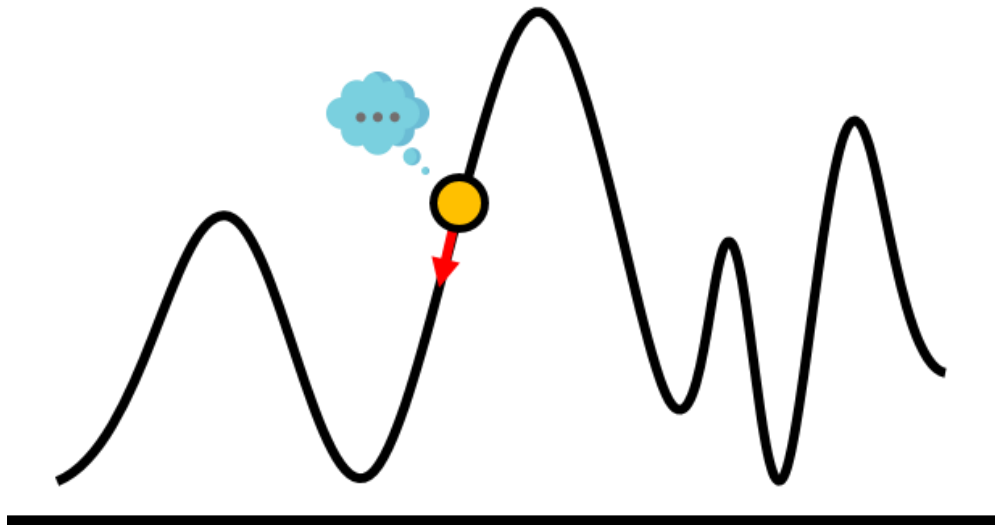
# 1. Introduction

## Swarm-Based Optimization

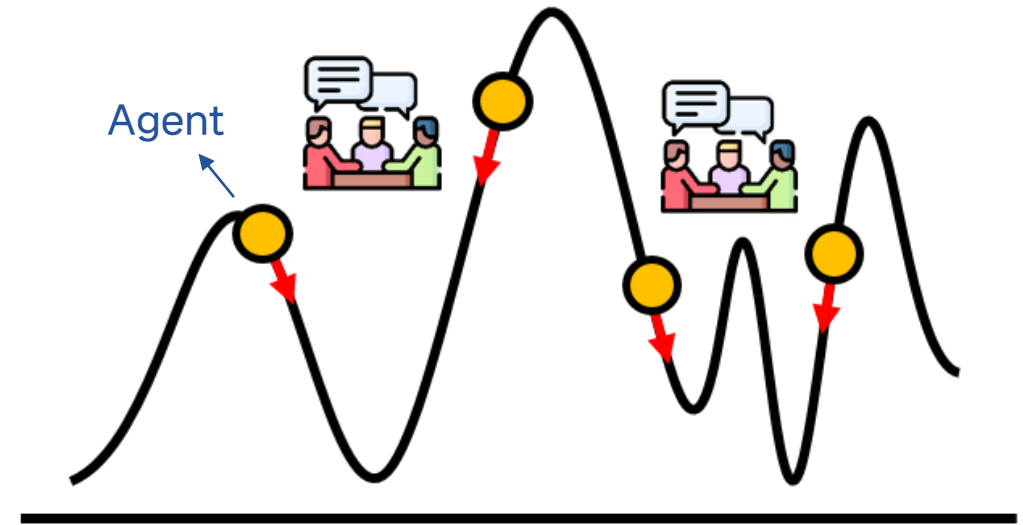
### Swarm-Based Optimization

- Blind Search, Local Search 와 같은 Single Agent Optimization과 달리 여러 개의 optimizer가 서로 정보를 교환하며 동시에 최적화를 수행

a. Single Agent Optimization



b. Swarm-Based Optimization



# 1. Introduction

## Swarm-Based Optimization

### Swarm-Based Optimization

- Blind Search, Local Search 와 같은 Single Agent Optimization과 달리 여러 개의 optimizer가 서로 정보를 교환하며 동시에 최적화를 수행
- 주어진 법칙과, Agent 들이 저장하고 있는 정보를 조합하여 최적해를 찾음
  - 단점: 계산량이 많음
  - 장점: 하나의 agent가 local optimum에 빠져도, 전체적으로는 global optimum에 수렴

# 1. Introduction

## Particle Swarm Optimization

### Particle Swarm Optimization

- 새 무리와 물고기 떼와 같은 동물들의 군집적인 행동 양상을 모방
- GA의 교차 및 변이와 같은 진화 연산자 -> 입자라고 불리는 후보 솔루션 집단을 이용하여 최적화 문제 해결

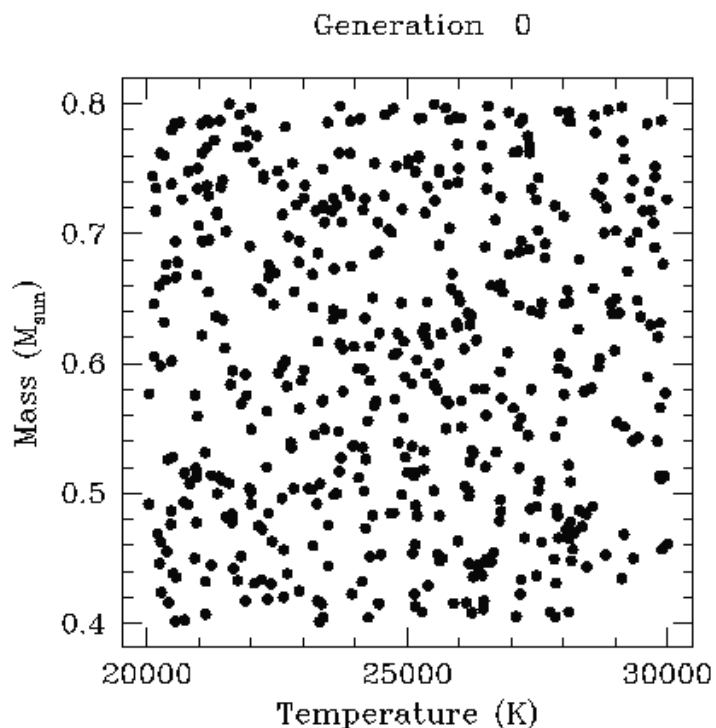


그림 1. GA를 이용한 최적점 탐색

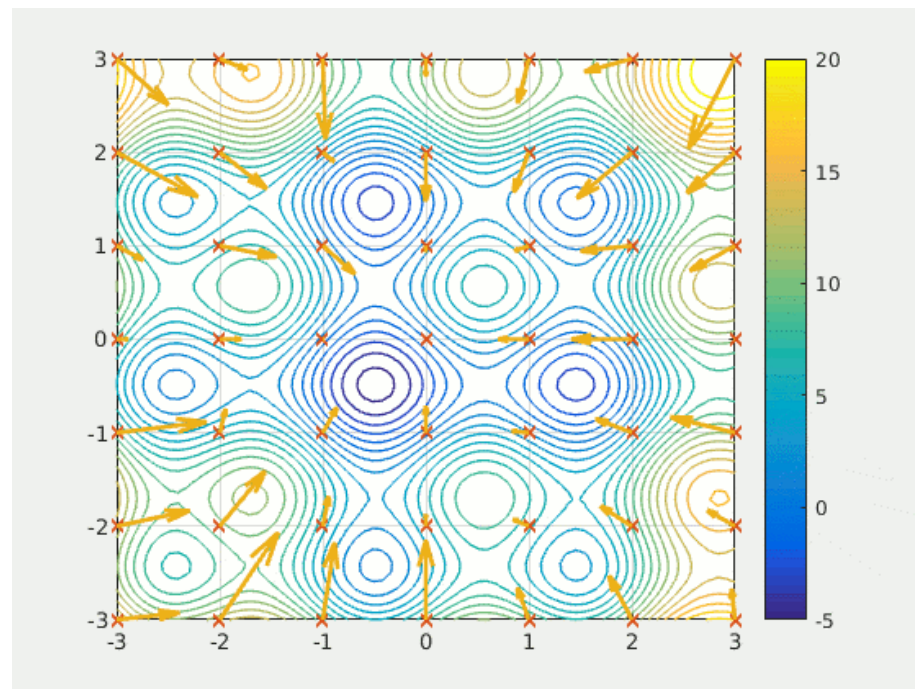


그림 2. PSO를 이용한 최적점 탐색

# 1. Introduction

## Particle Swarm Optimization

### Particle Swarm Optimization



관성력 : 이전 단계에서 속도

인지력 : 각 개체가 이동했던 지점 중 최상의 위치와 현재 위치의 거리 차

사회력 : 집단 중에서 최상의 위치와 현재 위치의 거리 차

## 리더를 따르자!



[따르기 위해 필요한 요소]

1. 리더의 위치
2. 리더에게 가기 위한 속도

## 2. Method

### Particle Swarm Optimization

### Particle Swarm Optimization

---

**Algorithm 7** Particle swarm optimization pseudo-code for SPSO 2007 and 2011

---

```
1: Inputs:  $f, C$  ▷  $f$  is the evaluation function,  $C$  includes control parameters
2:  $P \leftarrow \text{initialization}(C)$  ▷ set initial swarm (topology, random position and velocity,
   previous best and previous best position found in the neighborhood)
3:  $B \leftarrow \text{best}(P, f)$  ▷ best particle
4:  $i \leftarrow 0$  ▷  $i$  is the number of iterations of the method
5: while not  $\text{termination\_criteria}(P, f, C, i)$  do
6:   for each particle  $x = (s, v, p, l) \in P$  do ▷ cycle all particles
7:      $v \leftarrow \text{velocity}(s, v, p, l)$  ▷ compute new velocity for  $x$ 
8:      $s \leftarrow s + v$  ▷ move the particle to new position  $s$  (mutation)
9:      $s \leftarrow \text{confinement}(s, C)$  ▷ adjust position  $s$  if it is outside bounds
10:    if  $f(s) < f(p)$  then  $p \leftarrow s$  ▷ update previous best
11:    end if
12:     $x \leftarrow (s, v, p, l)$  ▷ update particle
13:    if  $f(s) < f(B)$  then  $B \leftarrow s$  ▷ update best value
14:    end if
15:  end for
16:   $i \leftarrow i + 1$ 
17: end while
18: Output:  $B$  ▷ best solution
```

---

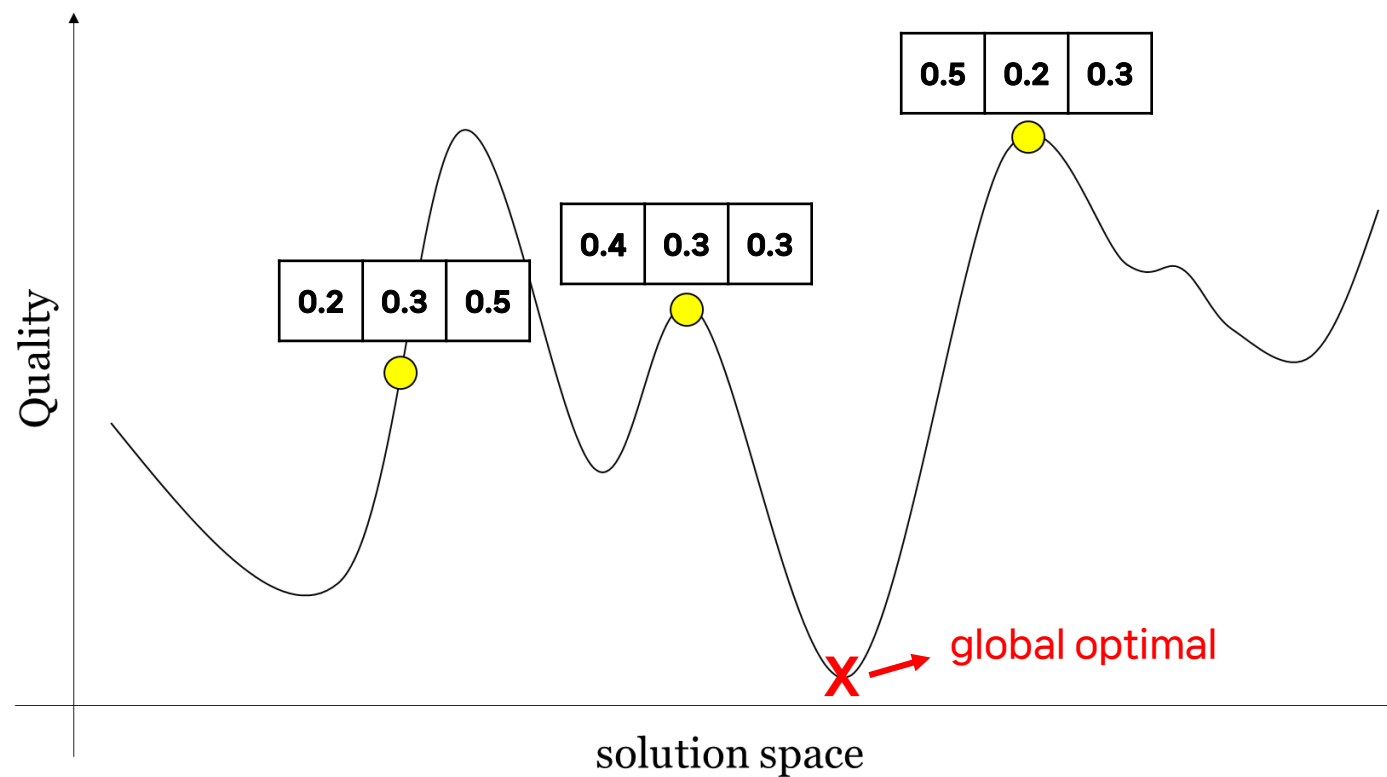
## 2. Method

### Particle Swarm Optimization

#### Learning Process

##### 1. Initialization

- 각 입자들의 위치와 속도를 초기화



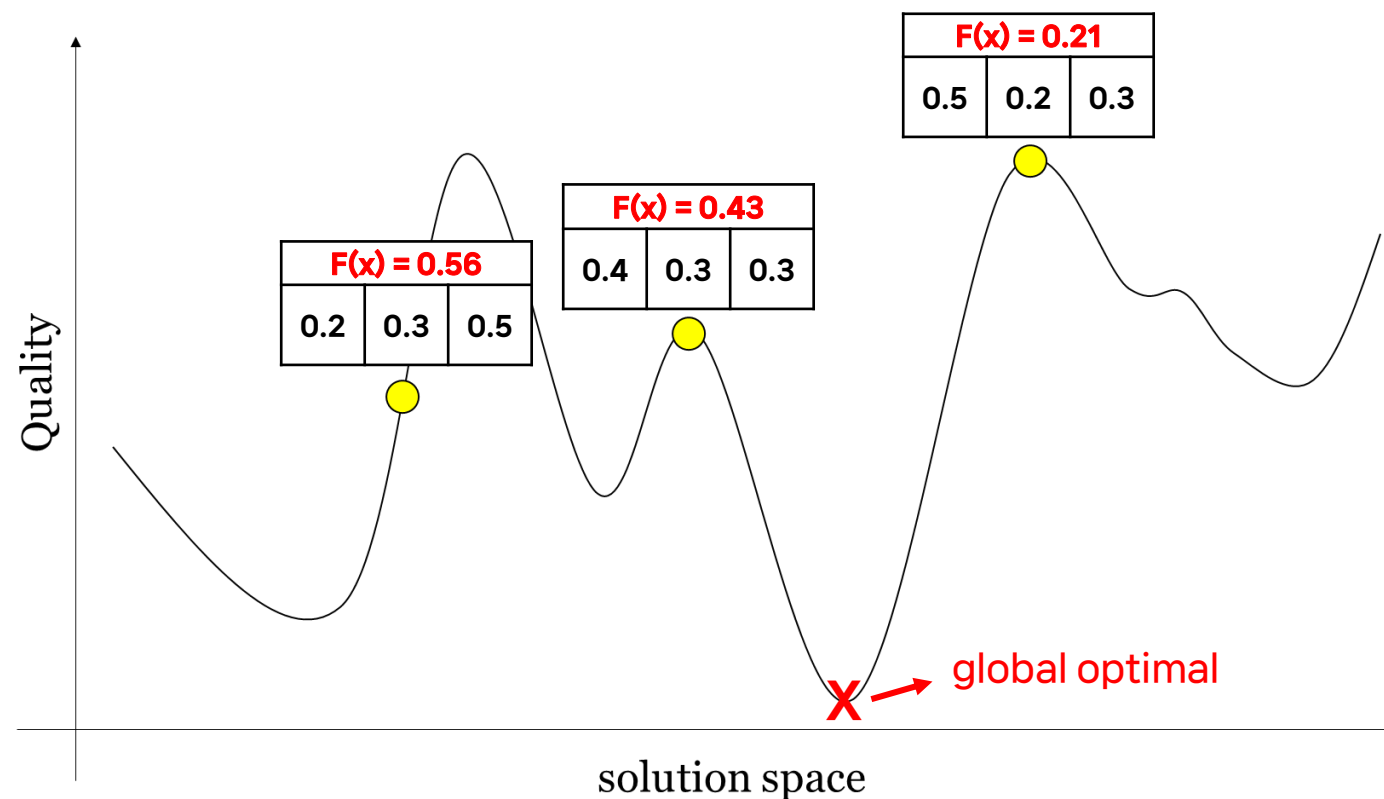
## 2. Method

### Particle Swarm Optimization

#### Learning Process

##### 2. Evaluation

- 입자의 위치를 평가하여 cost 계산





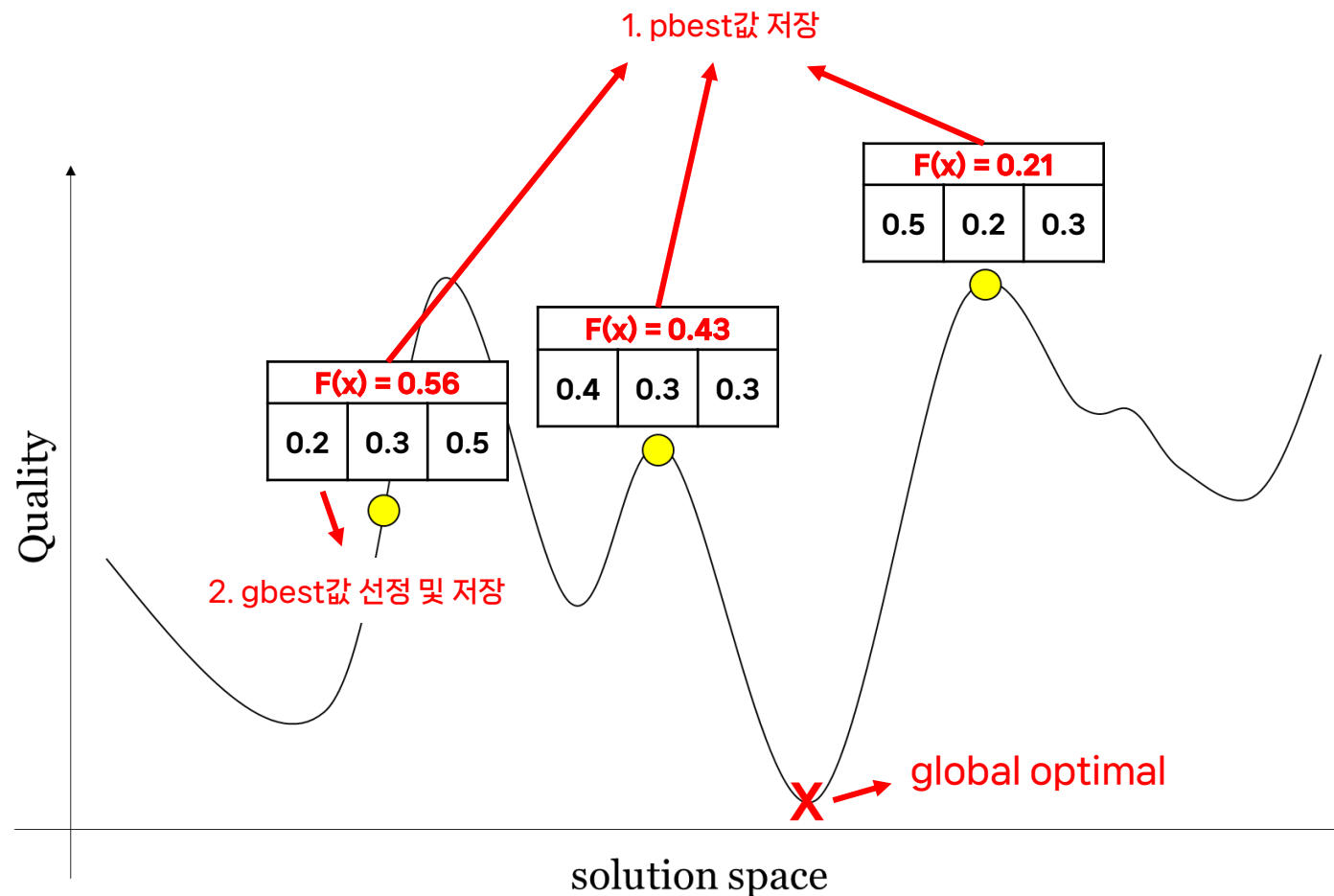
## 2. Method

### Particle Swarm Optimization

#### Learning Process

##### 3. Position Information Save

- 모든 입자는 경험한 최적의 위치를 저장
  - 각 입자의 최적 위치(pbest)
  - 군집의 최적 위치(gbest)



## 2. Method

### Particle Swarm Optimization

#### Learning Process

#### 4. Position & Velocity Update

- 속도 업데이트

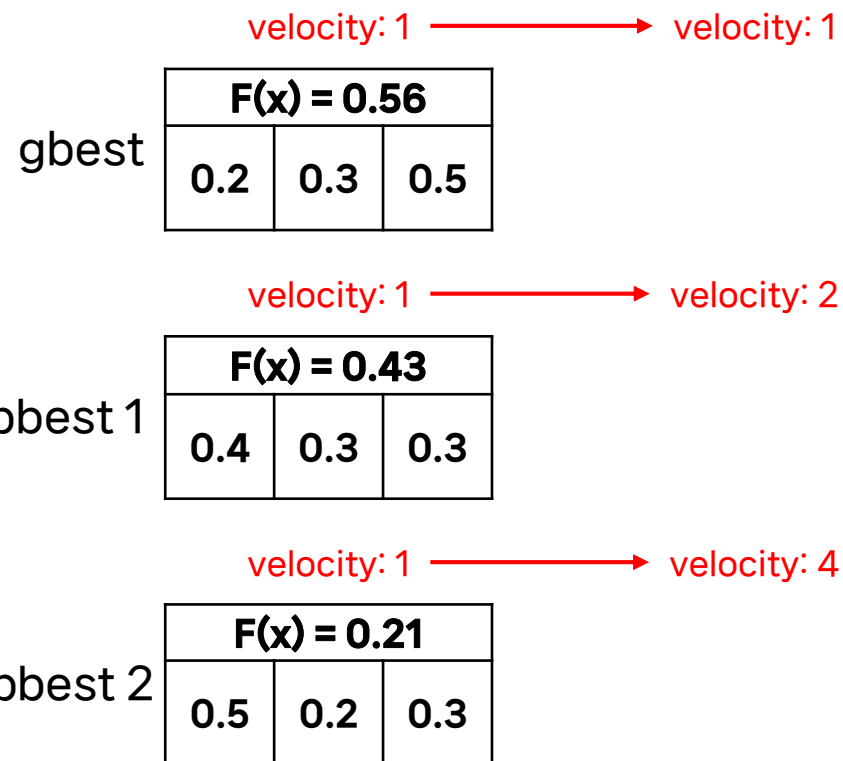
$$v_i = w \cdot v_i + r_1 \cdot (pbest_i - x_i) + r_2 \cdot (gbest_i - x_i)$$

$x_i$  : 입자의 위치

$w$  : 관성 가중치(입자가 이전 방향으로 일정하게 이동하려는 경향성을 반영하기 위해)

$r_1$  : 각 입자 정보의 중요도

$r_2$  : 군집 정보의 중요도



## 2. Method

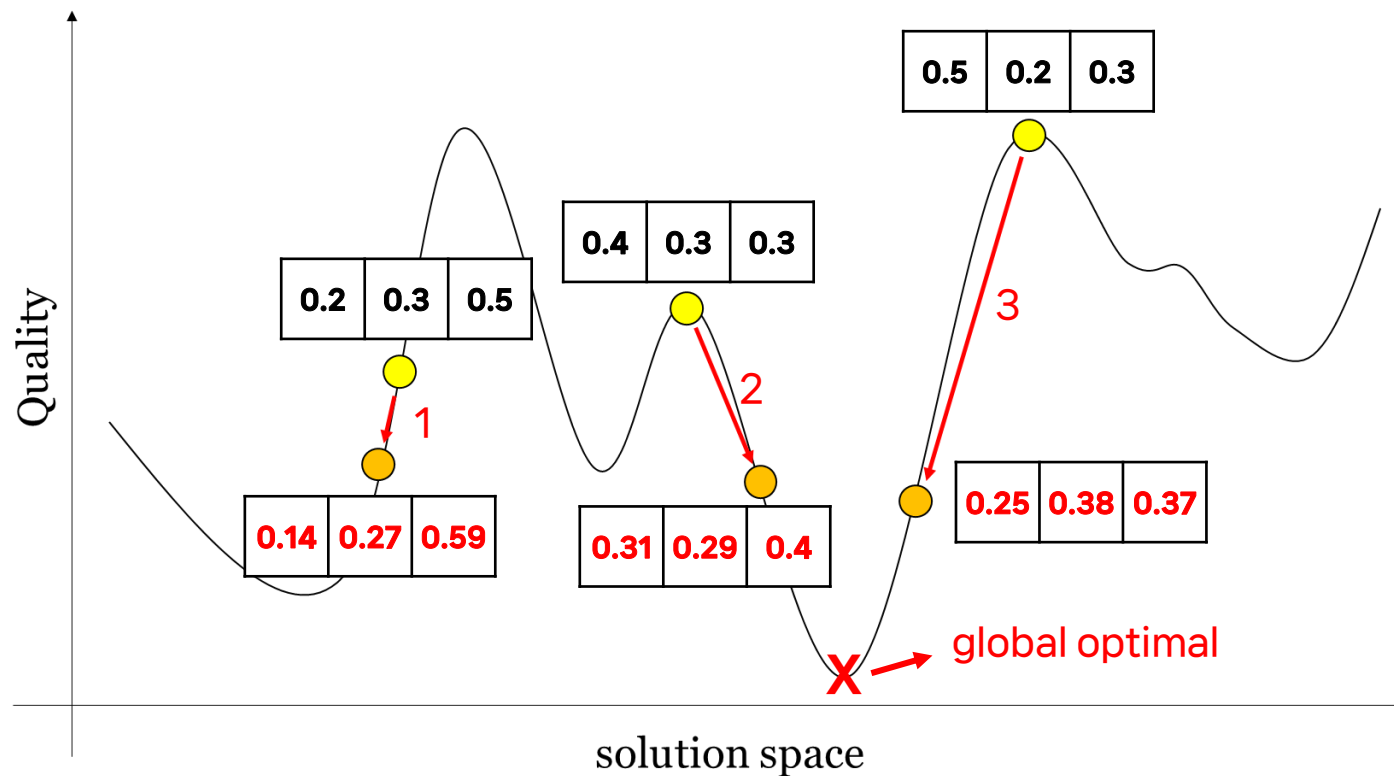
### Particle Swarm Optimization

#### Learning Process

##### 4. Position & Velocity Update

- 위치 업데이트

$$x_i = x_i + v_i$$



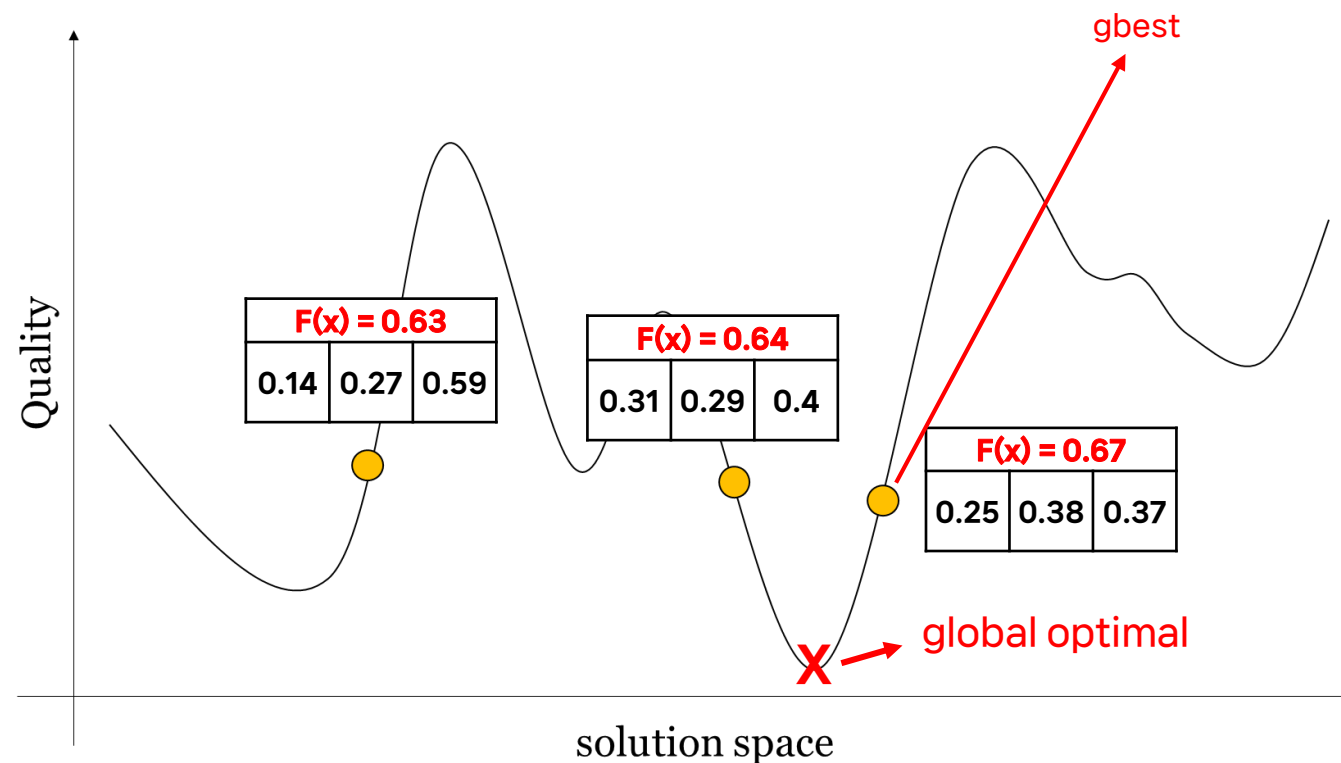
## 2. Method

### Particle Swarm Optimization

#### Learning Process

##### 5. Re-Evaluation -> 종료 or 반복

- 종료 조건
- max iteration / 특정 조건 만족



## 2. Method

### Particle Swarm Optimization

#### Learning Process

Generation 1

gbest

<b>F(x) = 0.56</b>			<b>1</b>
0.2	0.3	0.5	

Generation 2

pbest

<b>F(x) = 0.63</b>			<b>1</b>
0.14	0.27	0.59	

Generation 3

pbest

<b>F(x) = 0.70</b>			<b>1.5</b>
0.39	0.29	0.42	

Generation N

pbest

<b>F(x) = 0.43</b>			<b>1</b>
0.4	0.3	0.3	

pbest

<b>F(x) = 0.64</b>			<b>2</b>
0.31	0.29	0.4	

gbest

<b>F(x) = 0.82</b>			<b>1.5</b>
0.42	0.33	0.25	

...

<b>F(x) = 0.98</b>			<b>0.01</b>
0.31	0.32	0.37	

pbest

<b>F(x) = 0.21</b>			<b>1</b>
0.5	0.2	0.3	

gbest

<b>F(x) = 0.67</b>			<b>4</b>
0.25	0.38	0.37	

pbest

<b>F(x) = 0.79</b>			<b>1.4</b>
0.35	0.33	0.42	

Solution

<b>F(x) = 0.99</b>			<b>0.01</b>
0.32	0.32	0.36	

## 3. Result

### Particle Swarm Optimization

#### GA와의 공통점

- 무작위로 생성된 집단으로 시작
- 모집단을 업데이트하고 무작위 기법으로 최적의 위치를 탐색
- Local minimum에 빠질 가능성이 낮으며 Global minimum을 찾을 수 있음

#### GA와의 차이점

- 입자 자신의 이동 속도를 스스로 업데이트
- 주변 입자와의 정보 교환을 통해 최적 위치를 기록
- PSO는 GA보다 계산량이 적음 -> 간단하기 때문에 복잡한 문제에서 자주 사용
- GA에서는 염색체가 서로 정보를 공유함으로 인해, 집단 전체는 하나의 그룹처럼 최적 방향으로 이동
- 각 입자가 개별 학습하여 Local minimum에서 빠져도, 신속하게 Global minimum으로 수렴할 수 있음

# Thanks