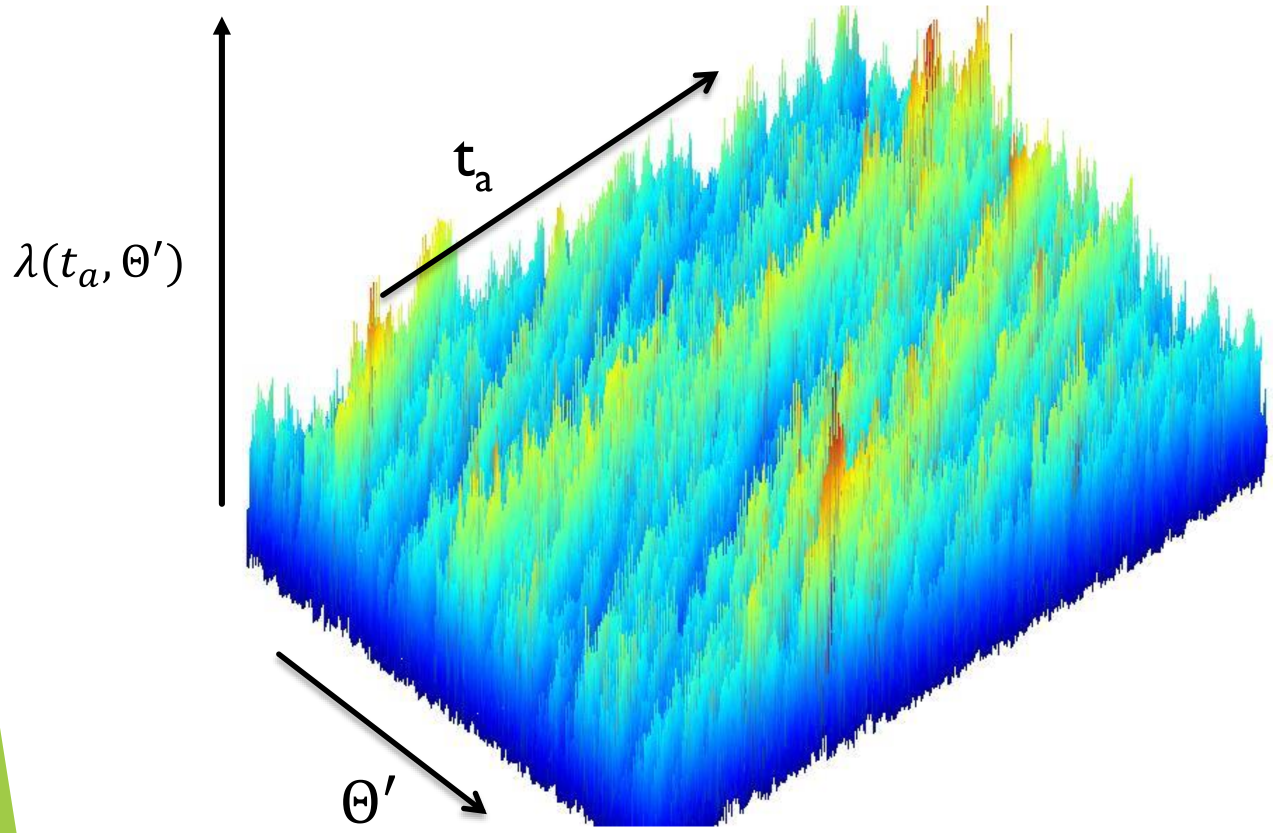
# GLRT optimization

Gravitational Wave Data Analysis School in China

Soumya D. Mohanty

UTRGV

# Binary inspiral search



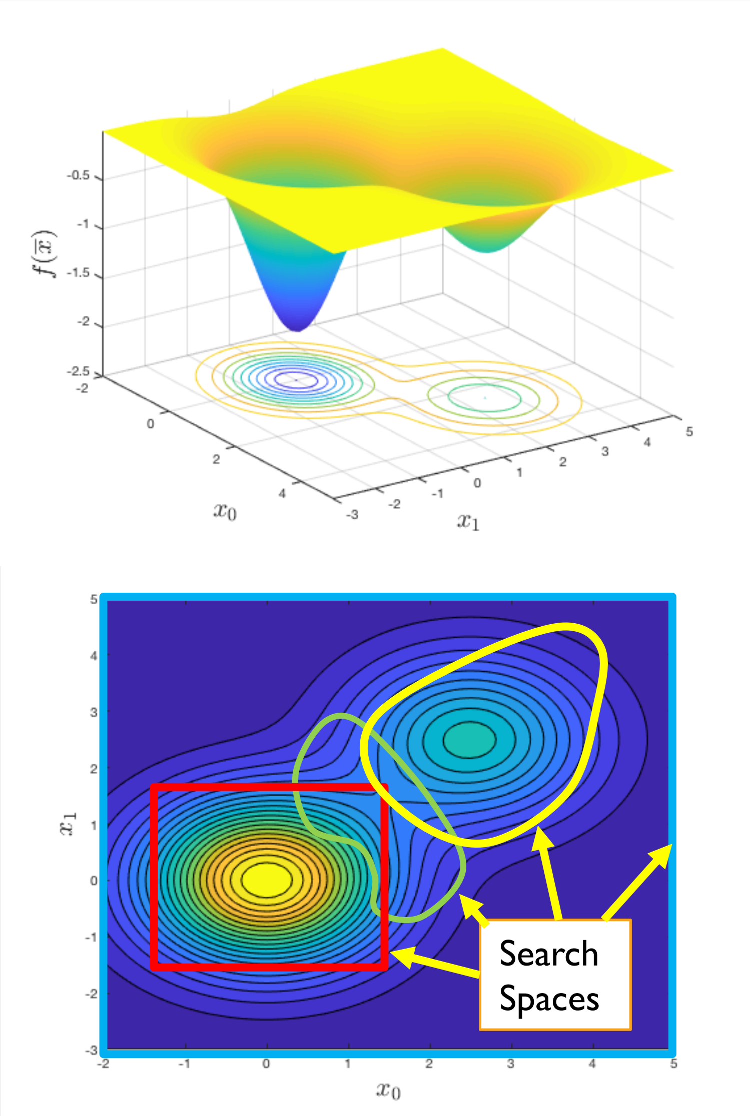

The numerical optimization problem is

1. Intrinsically difficult
   - Large number of maxima
   - Becomes worse as the number of parameters increases
2. Computationally expensive
   - Binary inspiral network analysis for ground-based detectors grid based search: $\approx 10^8$ points in $\Theta_{intrinsic}$ space with $\approx 10^7$ floating point operations per point (1 hour segments) $\Rightarrow$ 0.3 Tflops to just keep up with the incoming data rate
   - Computational bottleneck $\Rightarrow$ current searches follow a sub-optimal approach $\Rightarrow$ Lower sensitivity $\Rightarrow$ Reduced rate of detections

# GLRT optimization approaches

| Grid-based search | Stochastic search |
| --- | --- |
| • High computational cost<br>• forces the use of sub-optimal methods<br>• Examples:<br>  • Binary inspiral network analysis: network GLRT used only when single detector GLRT+MLE find signals that are coincident in time<br>  • Search for continuous waves from deformed pulsars: Hierarchical search strategy where the first step is just GLRT over 1 day intervals (GLRT required over 1 year data!) | • Random walk in parameter space<br>• Markov Chain Monte Carlo (MCMC): still very expensive<br>• **Particle Swarm Optimization** (PSO): Proving to be cheap and effective |

# Optimization terminology




# OPTIMIZATION: OBJECTIVE

- Continuous optimization problem: Find the minimum value of a function $f(\bar{x})$ in a specified domain $\bar{x} \in \mathbb{D} \subseteq \mathbb{R}^D$
- Minimizer: The location, $\bar{x}^*$, of the minimum

$$f(\bar{x}^*) \leq f(\bar{x}), \bar{x}^* \in \mathbb{D}, \bar{x} \in \mathbb{D}$$

- Alternatively

$$f(\bar{x}^*) = \min_{\bar{x} \in \mathbb{D}} f(\bar{x})$$
$$\bar{x}^* = \operatorname{argmin}_{\bar{x} \in \mathbb{D}} f(\bar{x})$$

- $f(\bar{x})$ is called the fitness function (also objective function)
- $\mathbb{D}$ is called the search space (or constraint set)
- *Maximization of $f(\bar{x})$ is equivalent to minimization of $-f(\bar{x})$

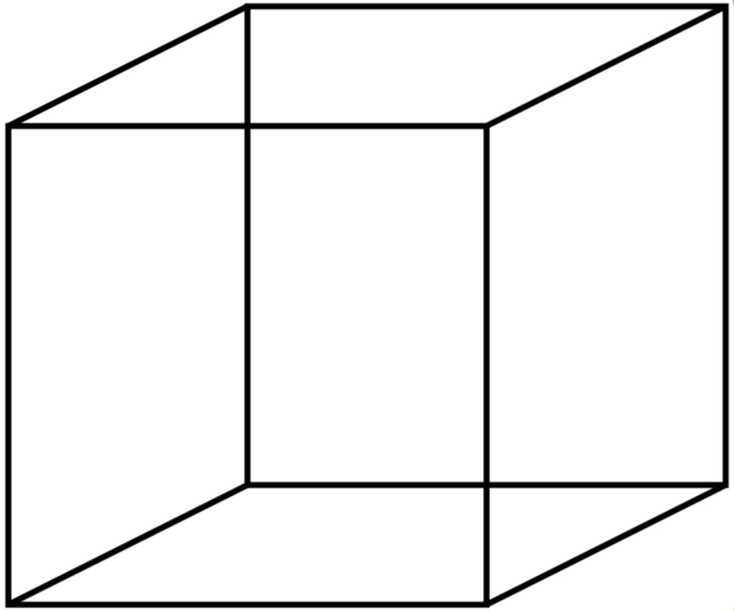
# GLRT and optimization terminology

- $L_G = \max_{\Theta} \langle \bar{y}, \bar{q}(\Theta\ ) \rangle^2$
  - Fitness function is: $-\langle \bar{y}, \bar{q}(\Theta\ ) \rangle^2$ (Because we only consider minimization)
  - Search space is the space of parameters $\Theta = (\theta_1, \theta_2, \dots, \theta_D)$ left after amplitude maximization
- $\mathrm{L}_G = \max_{\Theta} [\langle \bar{y}, \bar{q}_0(\Theta) \rangle^2 + \langle \bar{y}, \bar{q}_1(\Theta) \rangle^2]$
  - Fitness function is : $-(\langle \bar{y}, \bar{q}_0(\Theta) \rangle^2 + \langle \bar{y}, \bar{q}_1(\Theta) \rangle^2)$
  - Search space is the space of parameters $\Theta = (\theta_1, \theta_2, \dots, \theta_{D-1})$ left after amplitude and initial phase

# Standardized coordinates

**Hypercube search space:**

- If each parameter can be varied independently in some interval: $\theta_i \in [a_i, b_i]$
- **Caution**: Parameters may not be independent in some problems!





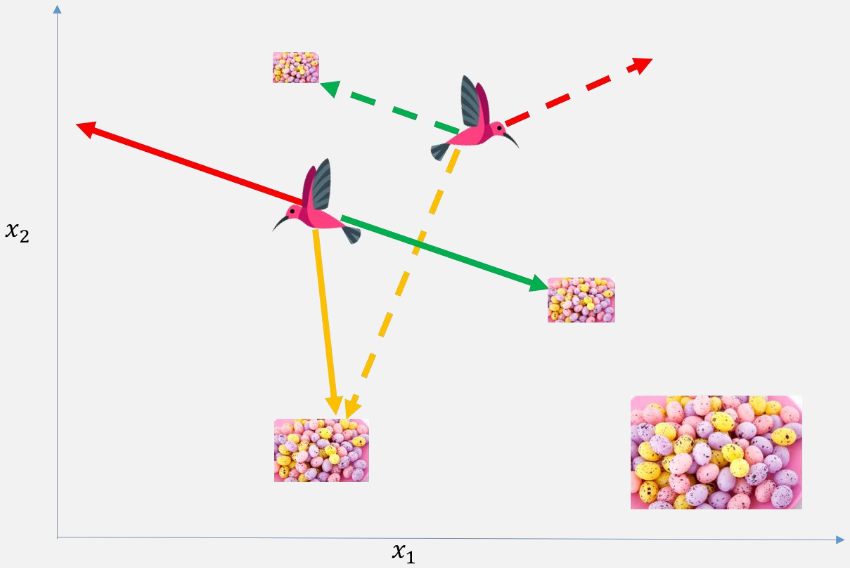
**Standardized coordinates:**

$$\theta_i \to x_i = \frac{\theta_i - a_i}{b_i - a_i}$$

$0 \leq x_i \leq 1$ for $\theta_i \in [a_i, b_i]$

# Particle swarm optimization

Selected topics from Chapters 4 and 5 of textbook



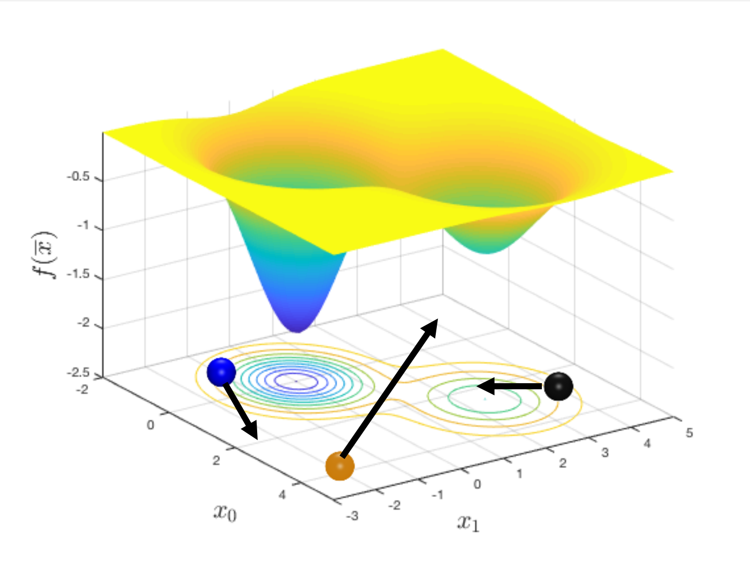

# PARTICLE SWARM OPTIMIZATION

- A swarm intelligence method inspired by the flocking behavior of birds
- Each agent moves under random attraction towards the best food sources that it and the swarm have found

# PARTICLE SWARM OPTIMIZATION

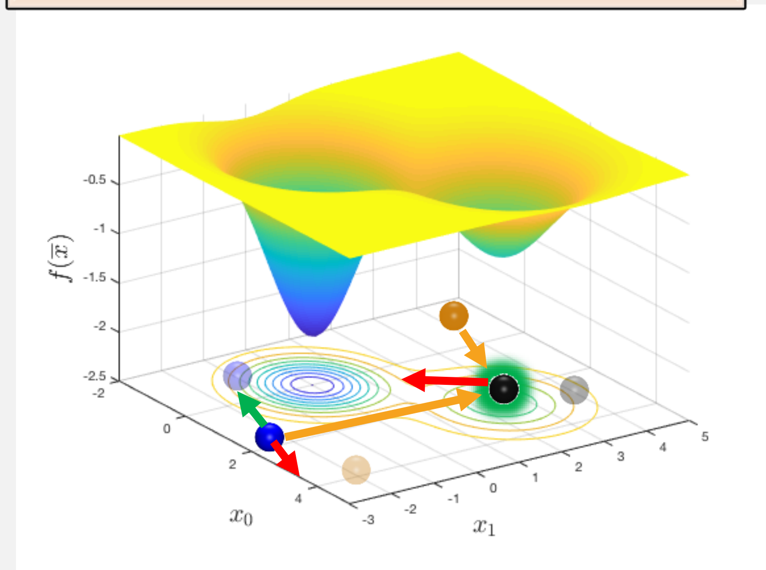
## Initialization

- Particle: agent location
- Particle fitness: Fitness value at location
- Particle "velocity": Displacement vector to new position



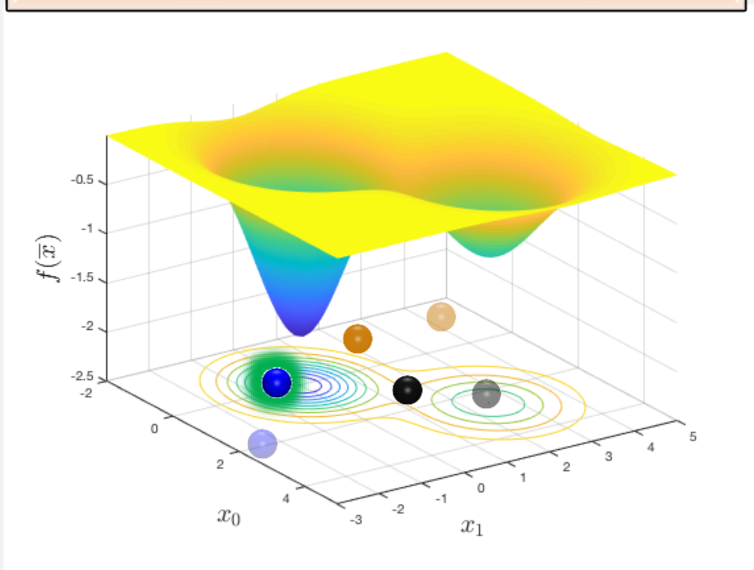

## Velocity update

- New velocity: sum of old velocity + acceleration terms
- Acceleration strengths are random




## Position update

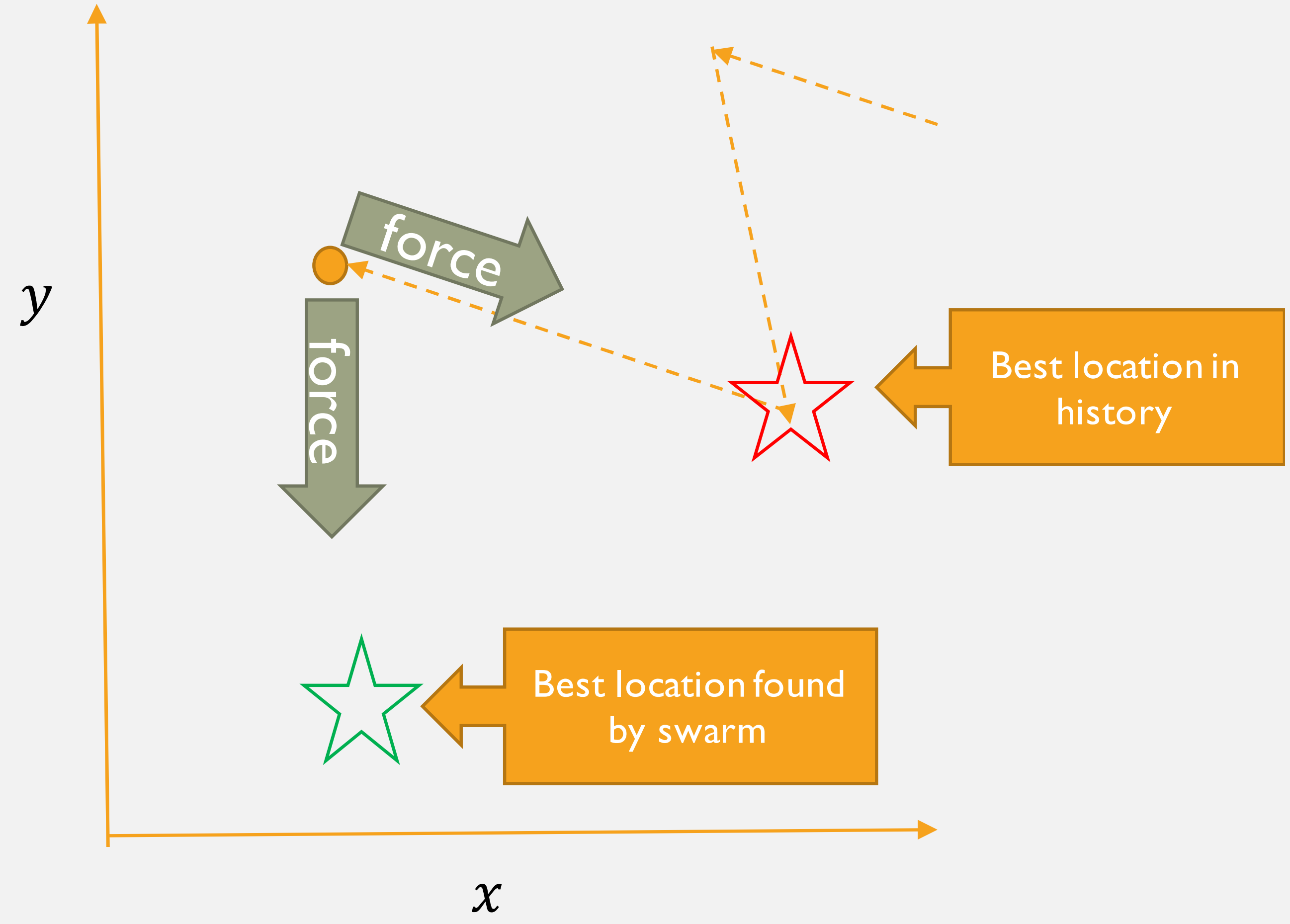
- Particles move to new positions



# VELOCITY UPDATE

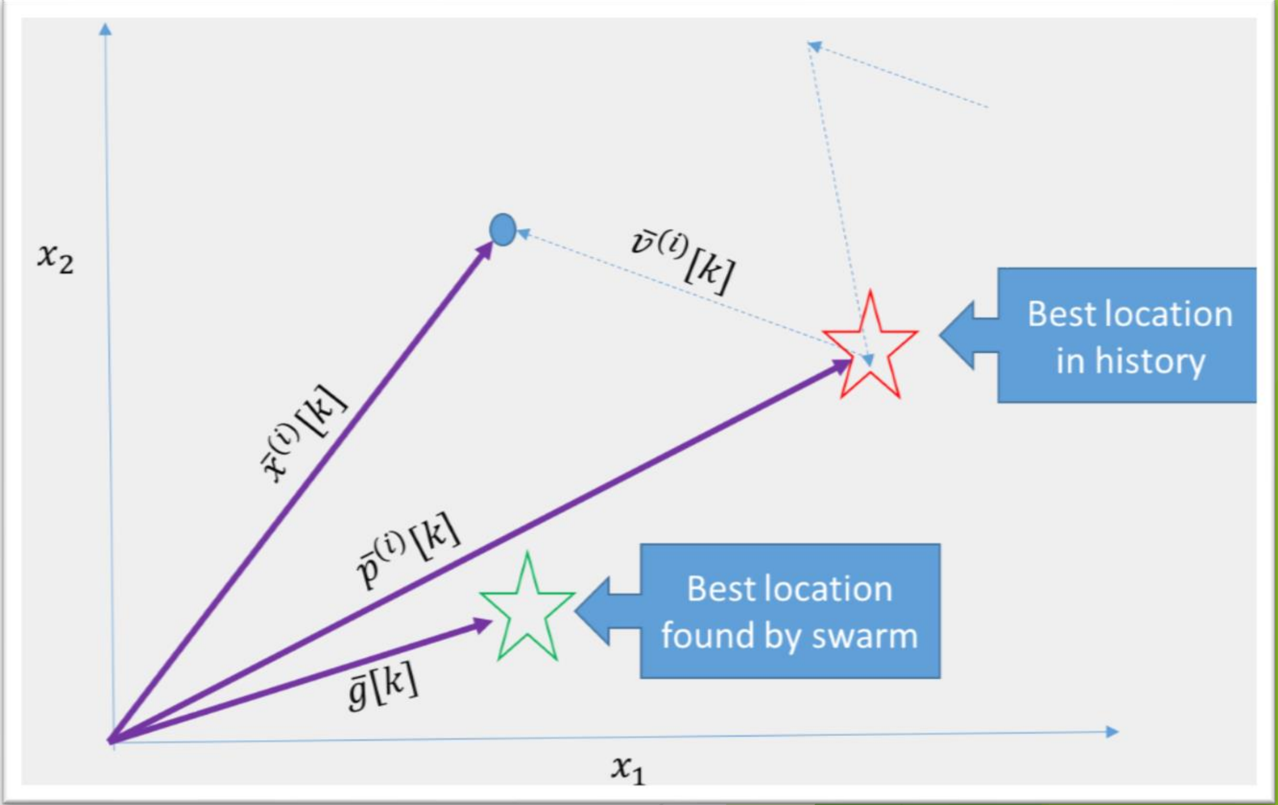
A particle explores the search space randomly but constantly feels an attractive force towards:

1. Personal best: best location it has found so far and …
2. Global best: the best location found by the swarm so far



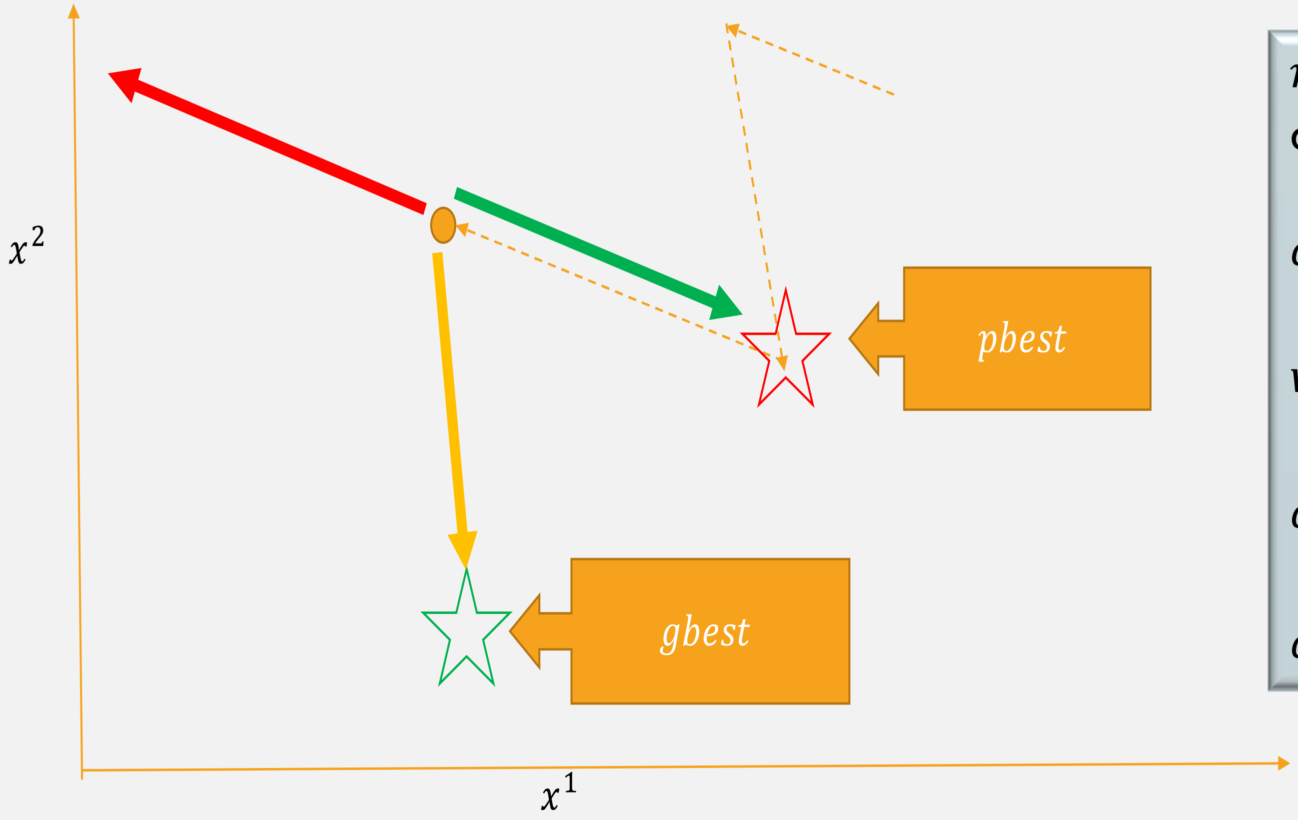
# PSO terminology

| Term | Definition |
|---|---|
| Particles | Locations in $D$-dimensional search space |
| $\bar{x}^{(i)}[k]$ | • Position of $i^{th}$ particle in $k^{th}$ iteration<br>• $\bar{x}^{(i)}[k] = (x_0^{(i)}[k], x_1^{(i)}[k], \dots, x_D^{(i)}[k])$ |
| $\bar{v}^{(i)}[k]$ | • Velocity of $i^{th}$ particle in $k^{th}$ iteration<br>• $\bar{v}^{(i)}[k] = (v_0^{(i)}[k], v_1^{(i)}[k], \dots, v_D^{(i)}[k])$ |
| $pbest$ ($\bar{p}^{(i)}[k]$) | Best location found by the $i^{th}$ particle over iterations 1 through $k$ |
| $gbest$ ($\bar{g}[k]$) | Best location found by any particle over iterations 1 through $k$ |
| $v_{max}$ | Maximum velocity<br>"Velocity Clamping": $v_j^{(i)}[k] \in [-v_{max}, v_{max}]$ |



# VELOCITY UPDATE

$$v_j^{(i)}[k+1] = w\, v_j^{(i)}[k] + c_1 r_{1,j}\left(p_j^{(i)}[k] - x_j^{(i)}[k]\right) + c_2 r_{2,j}\left(g_j\,[k] - x_j^{(i)}[k]\right)$$




$r_{m,j}$: random variable with uniform distribution in $[0,1]$

$c_1, c_2$ :"**acceleration constants**"

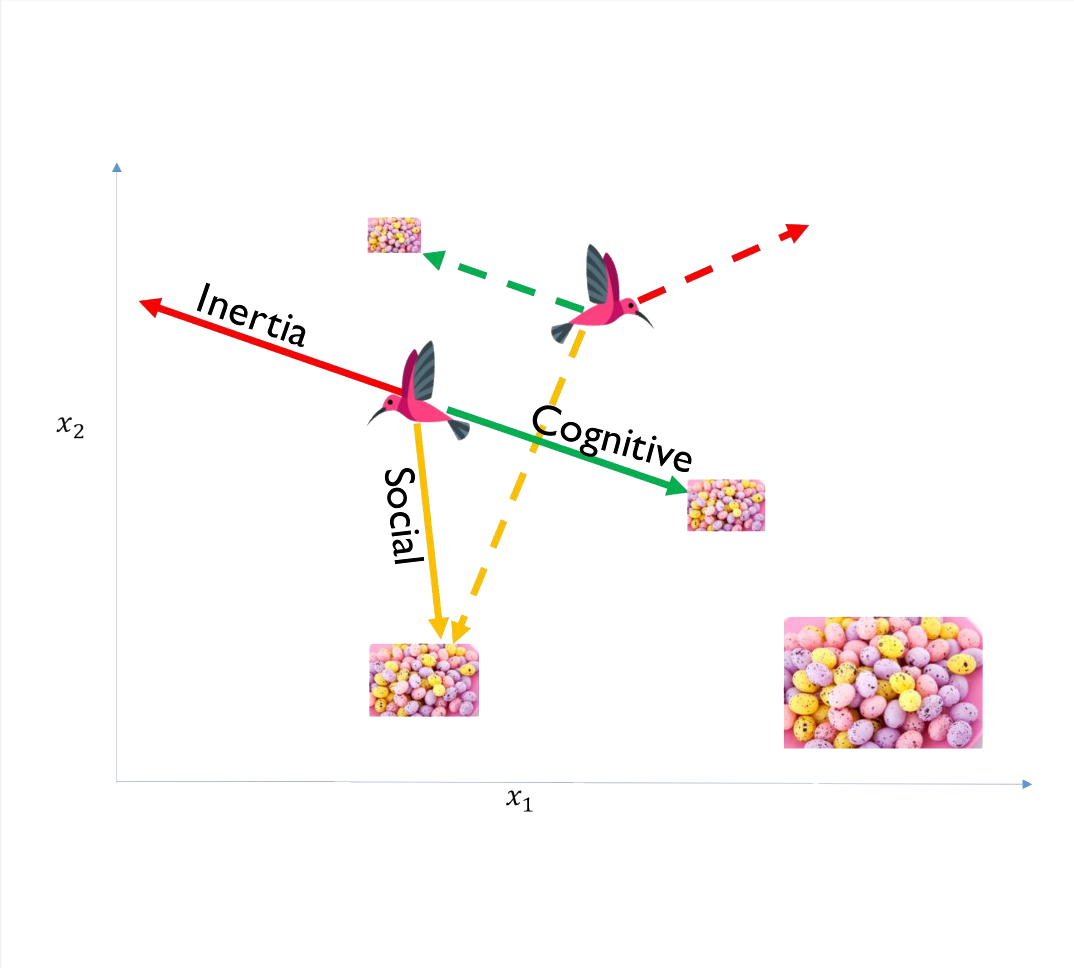
$w$ : "inertia" $\rightarrow w\, v_j^{(i)}[k]$ :"**Inertia Term**"

$c_1 r_{1,j}(p_i^j[k] - x_i^j[k])$ :"**Cognitive term**"

$c_2 r_{2,j}(g[k] - x_i^j[k])$ :"**Social term**"

# INTERPRETATION

- Inertia term: promotes exploration
  - $w < 1$ to avoid "particle explosion"
  - Common choice: Linear decay of $w$
- Social and cognitive terms: promote exploitation
  - Randomization in these terms promotes exploration



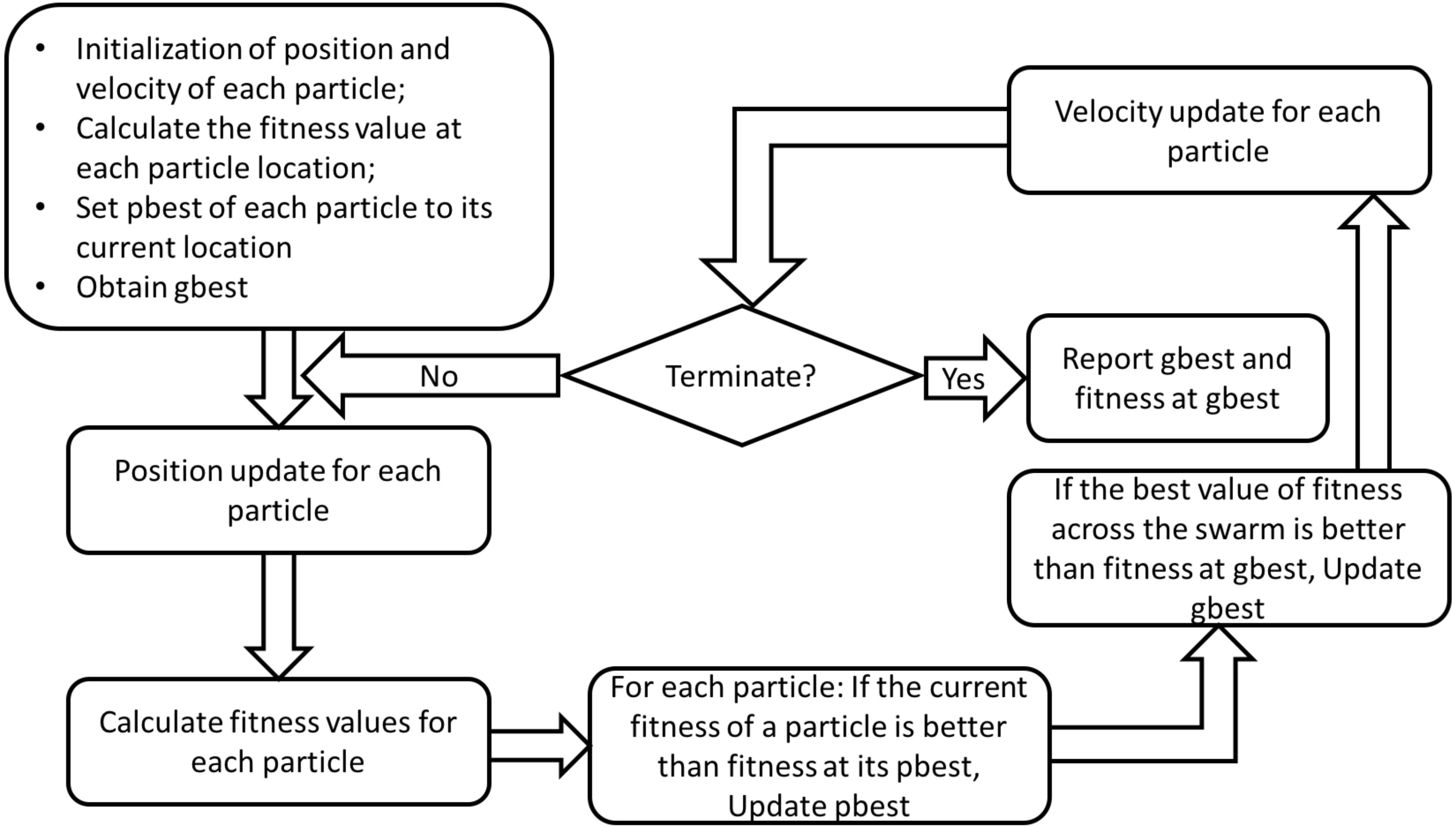
# PSO DYNAMICAL EQUATIONS

Velocity update

$$v_j^{(i)}[k+1] = w\, v_j^{(i)}[k] + c_1 r_{1,j}(p_j^{(i)}[k] - x_j^{(i)}[k]) + c_2 r_{2,j}(g_j\,[k] - x_j^{(i)}[k])$$

Position update

$$x_j^{(i)}[k+1] = x_j^{(i)}[k] + v_j^{(i)}[k+1]$$

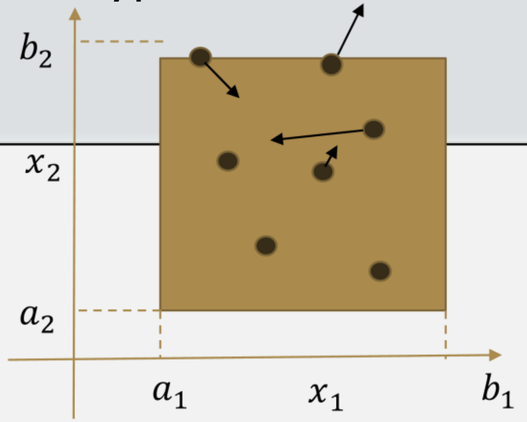


Initialization of position and velocity of each particle;
Calculate the fitness value at each particle location;
Set pbest of each particle to its current location
Obtain gbest
Velocity update for each particle
No
Terminate?
Yes
Report gbest and fitness at gbest
Position update for each particle
If the best value of fitness across the swarm is better than fitness at gbest, Update gbest
Calculate fitness values for each particle
For each particle: If the current fitness of a particle is better than fitness at its pbest, Update pbest

# INITIALIZATION AND TERMINATION

## Initialization

- $x_j^{(i)}[0]$ is picked from a uniform distribution $U(a_j, b_j)$ over $[a_j, b_j]$
- Search space assumed to be a hypercube




## Initial velocity

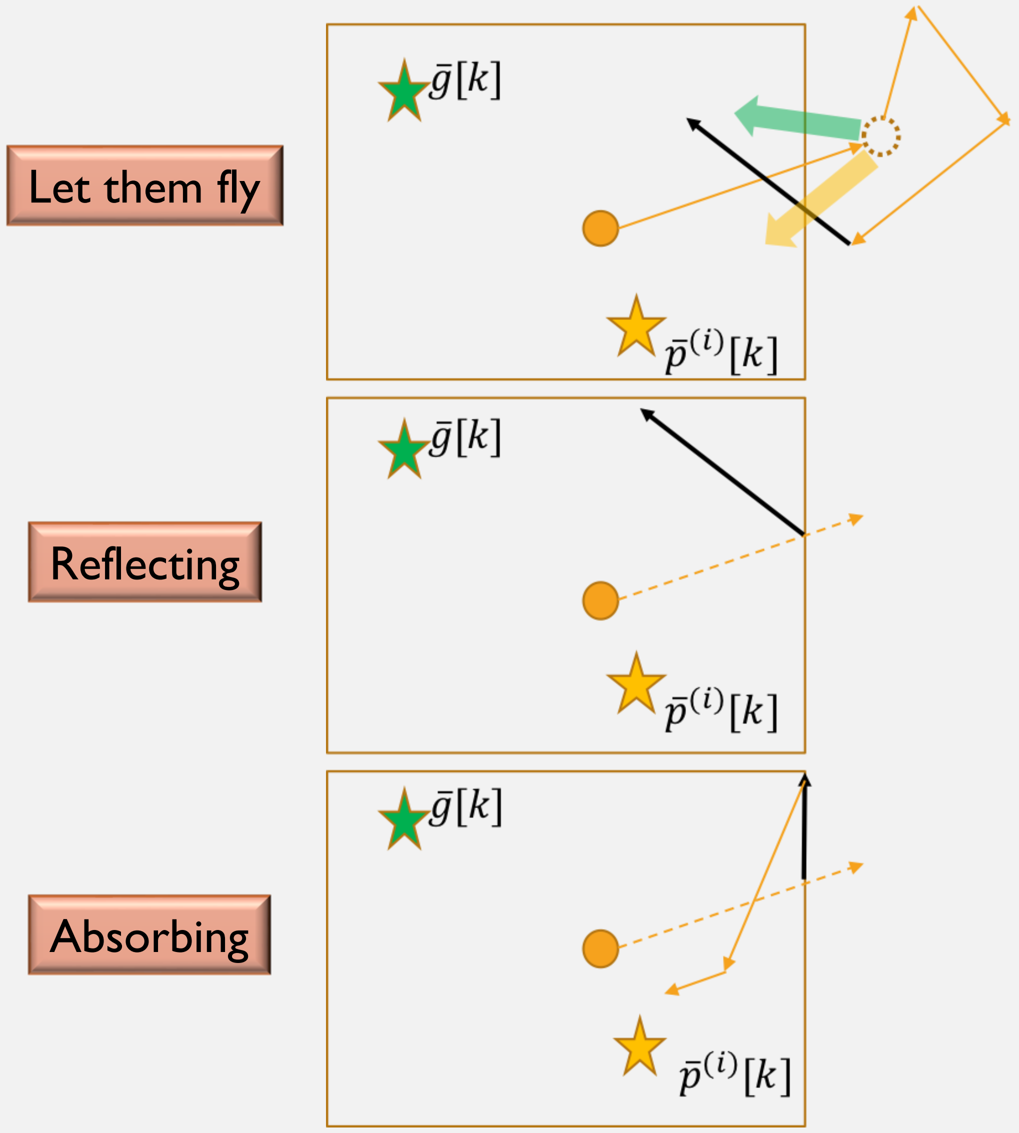
- Boundary constrained:
  - $v_j^{(i)}[0] \sim U(a_j - x_j^{(i)}[0], b_j - x_j^{(i)}[0])$ & velocity clamping

## Termination condition

- Number of iterations

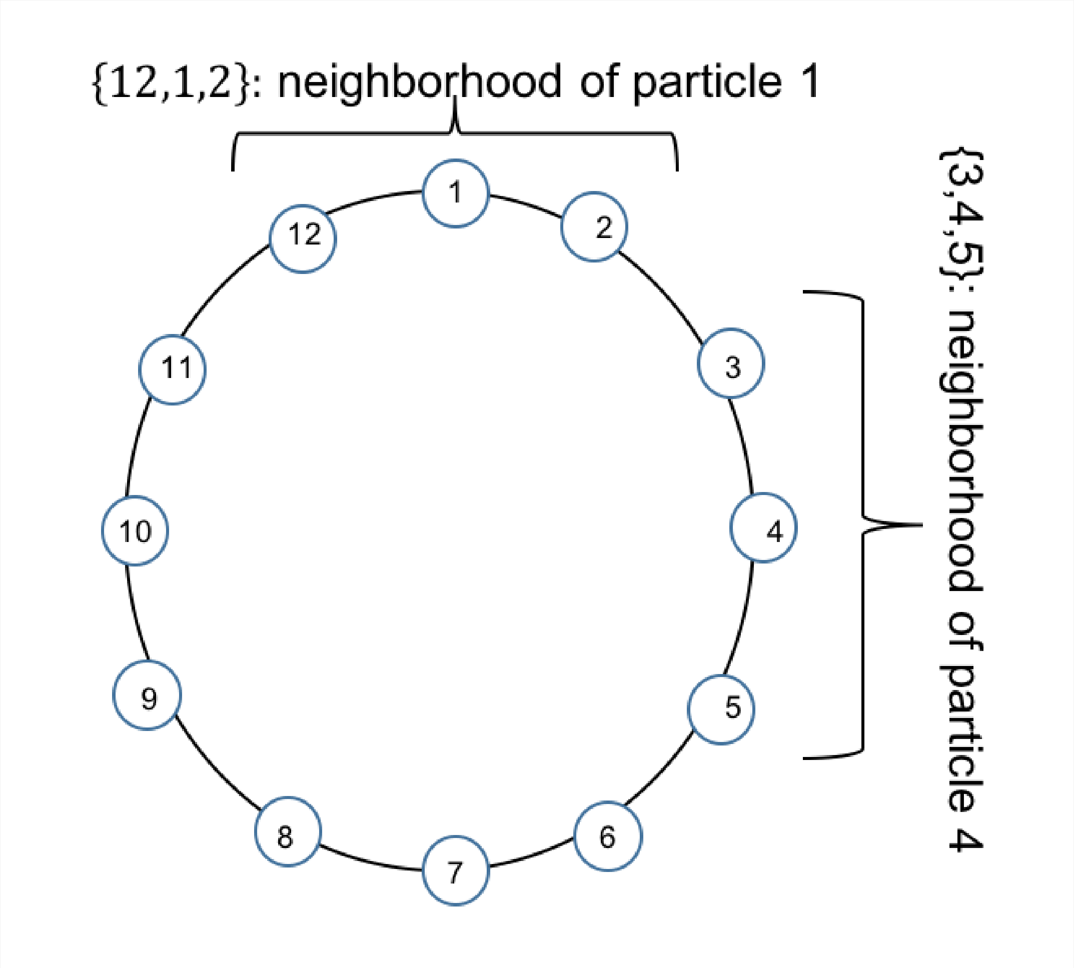
# BOUNDARY CONDITIONS

- "Let them fly": set fitness to $+\infty$ outside the boundary and continue to iterate the dynamical equations
  - $pbest$ and $gbest$ eventually pull the particle back
- "Reflecting walls" : Change the sign of the velocity component perpendicular to the boundary surface
- "Absorbing Walls": zero the velocity component perpendicular to the boundary surface



# PSO VARIANTS

See textbook for more discussion



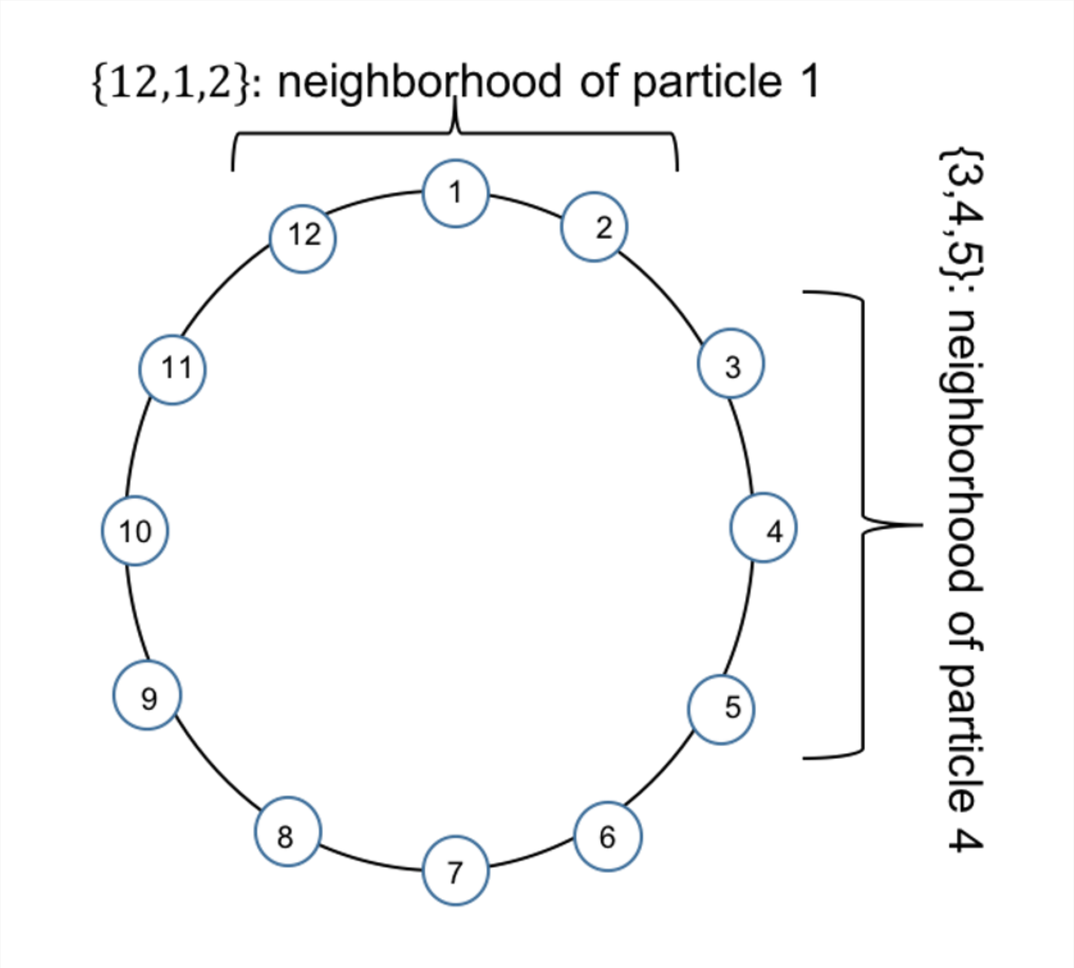

# COMMUNICATION TOPOLOGY

$$v_j^{(i)}[k+1] = w[k]v_j^{(i)}[k] + c_1 r_{1,j}\left(p_j^{(i)}[k] - x_j^{(i)}[k]\right) + c_2 r_{2,j}(g_j[k] - x_j^{(i)}[k])$$

<u>Local best PSO</u>

$\bar{g}[k] \rightarrow \bar{l}^{(i)}[k]$ : best value in a neighborhood of the $i^{\text{th}}$ particle

$$lbest: \bar{l}^{(i)}[k]$$




# $lbest$ PSO

Local best PSO

$$\bar{g}[k] \rightarrow \bar{l}^{(i)}[k]$$

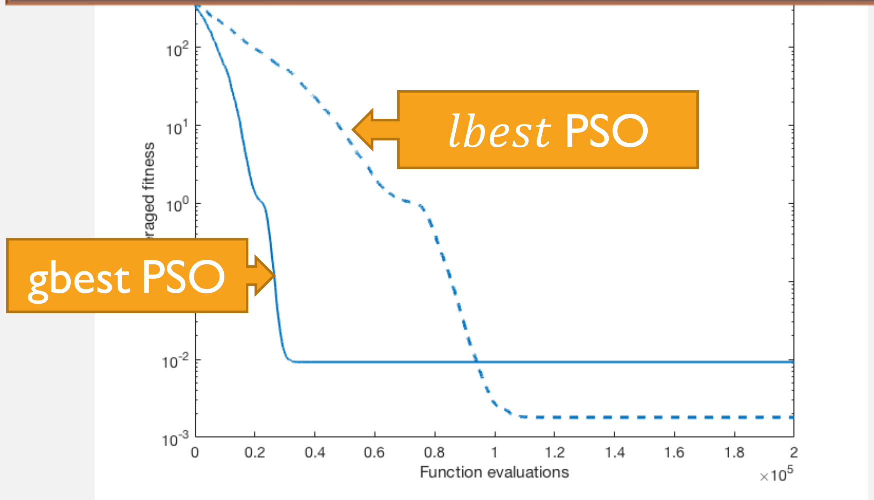
- Information about global best (e.g., particle #5) shared through common particles

$$\ldots, (1, 2, 3), (2, 3, 4)$$

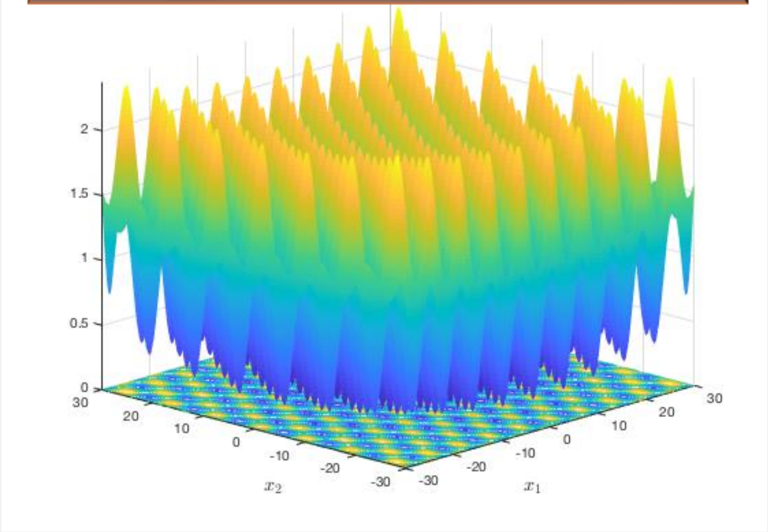
$$(2, 3, 4), (3, 4, 5), \ldots$$

- Information about global best propagates more slowly through the swarm
- Less social attraction: extended exploration

30D Generalized Griewank; $x_i \in [-600,600]$




2D Generalized Griewank




# *lbest* PSO PERFORMANCE

- Trade off between convergence and number of iterations: *lbest* PSO is computationally more expensive than *gbest* PSO

# RECOMMENDED PSO PARAMETER SETTINGS

- Follows Bratton and Kennedy, 2007
- Optimum particle number ($N_{part}$)
  - Too few $\Rightarrow$ Less exploration
  - Too many $\Rightarrow$ Premature convergence
- *lbest* PSO with ring topology (2 nearest neighbors)
  - Increases exploration
  - Slower convergence but often better probability of success

**Standardized coordinates**

| Setting Name | Setting Value |
|---|---|
| Position initialization | $x_j^{(i)}[0]$ drawn from $U(x;0,1)$ |
| Velocity initialization | $v_j^{(i)}[0]$ drawn from $U(x;0,1) - x_j^{(i)}[0]$ |
| $v_{\max}$ | 0.5 |
| $N_{\text{part}}$ | 40 |
| $c_1 = c_2$ | 2.0 |
| $w[k]$ | Linear decay from 0.9 to 0.4 |
| Boundary condition | Let them fly |
| Termination condition | Fixed number of iterations |
| lbest PSO | Ring topology; Neighborhood size $= 3$ |

# "BEST OF M RUNS" STRATEGY

probability of "success" in one run: $p$

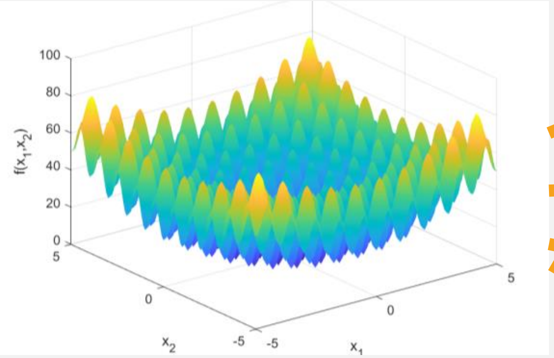
Probability of failure over M runs: $(1-p)^M$

- Example: If $p = 0.5$, failure probability over $M = 10$ runs is $\simeq 0.001$

Tuning strategy: Target a moderately high $p$ and pick best fitness from $M$ runs

- Moderate tuning reduces the danger of over-tuning
- Reduces the effort needed to achieve good tuning

# PRALLELIZATION IN BMR STRATEGY




Fitness function