SWARM INTELLIGENCE METHODS FOR STATISTICAL REGRESSION

Lecture 3

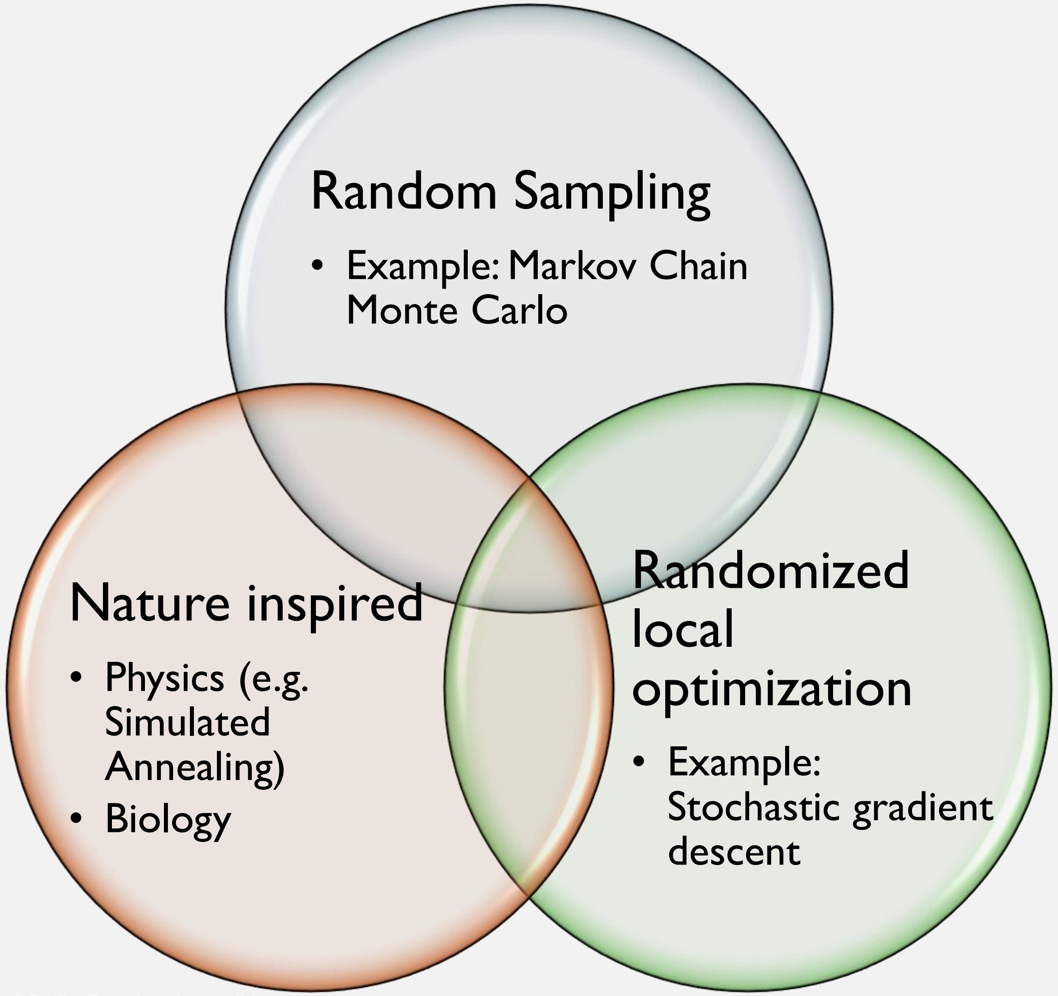
Soumya D. Mohanty

University of Texas Rio Grande Valley

STOCHASTIC OPTIMIZATION METHODS

Brief overview

METAHEURISTICS



Well-established approaches in stochastic optimization literature

The boundaries of the metaheuristics are permeable

New metaheuristics arise quite often

OPTIMIZATION IN BIOLOGICAL PHENOMENA

Multiple agents exploring a search space

+

Inter-agent communication

→

Objective: optimize some fitness metric

BIOLOGY INSPIRED METHODS

Evolutionary computation (EC)

Species fitness

- Natural evolution
- Communication: Sexual reproduction
- Example: Genetic Algorithm (GA)
- Agents die and are born

Swarm intelligence (SI)

Find food/avoid predators

- Flocking
- Communication: Monitor the fitness of neighbors
- Example: Particle swarm optimization (PSO)
- Agents do not die

Find food

- Foraging
- Communication: chemical trails
- Example: Ant colony optimization (ACO)
- Agents do not die

GENETIC ALGORITHM

Initialization

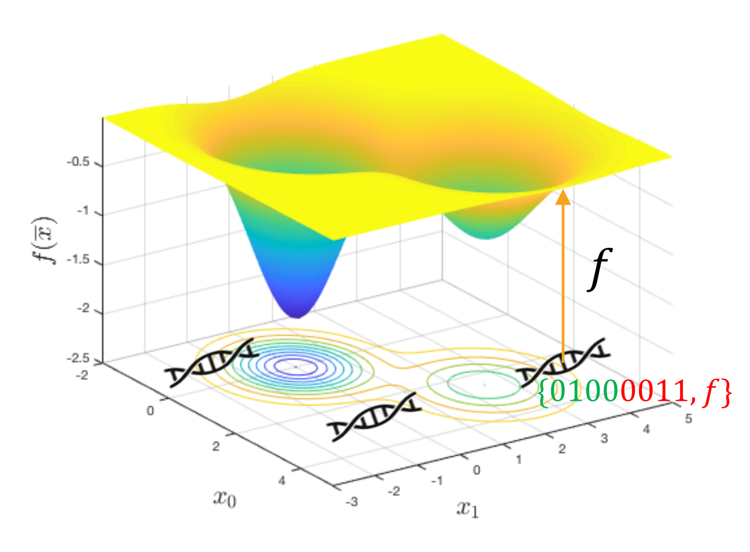
- Genome: Representation of agent location
- Genome fitness: Fitness function value at agent location

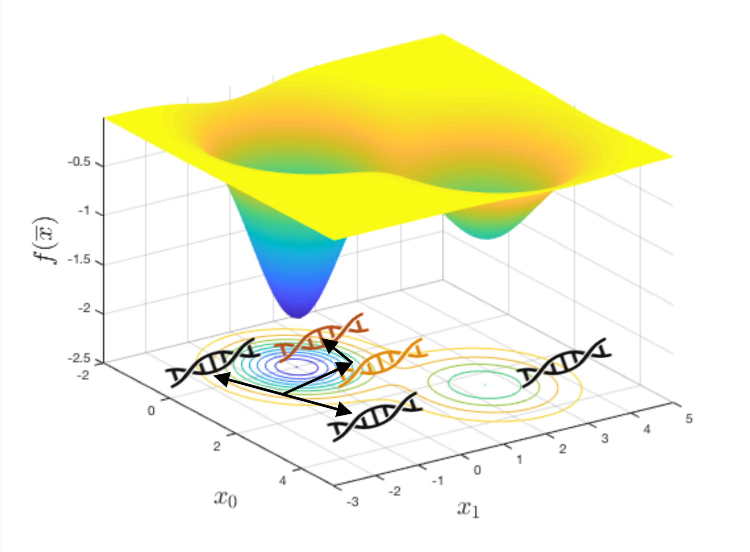
Crossover & Mutation

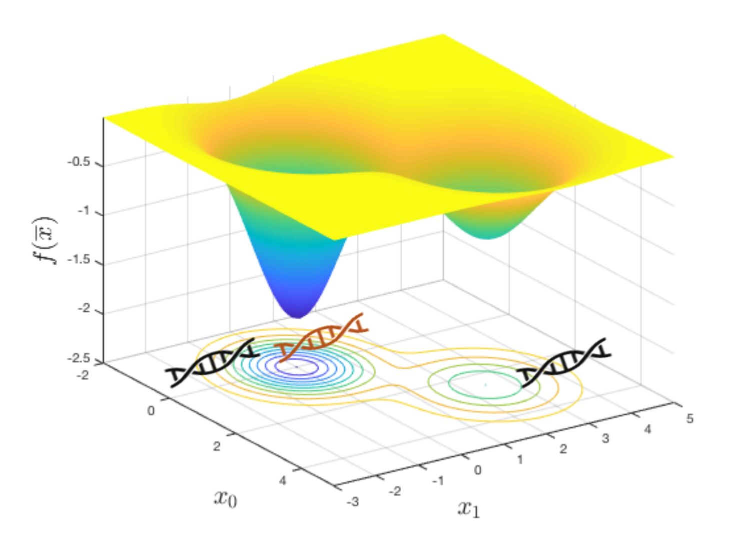
- Mix parent genomes to produce a child genome
- Random changes in child genome

Selection:

- Genomes with better fitness have higher chances of surviving







DIFFERENTIAL EVOLUTION

Initialization

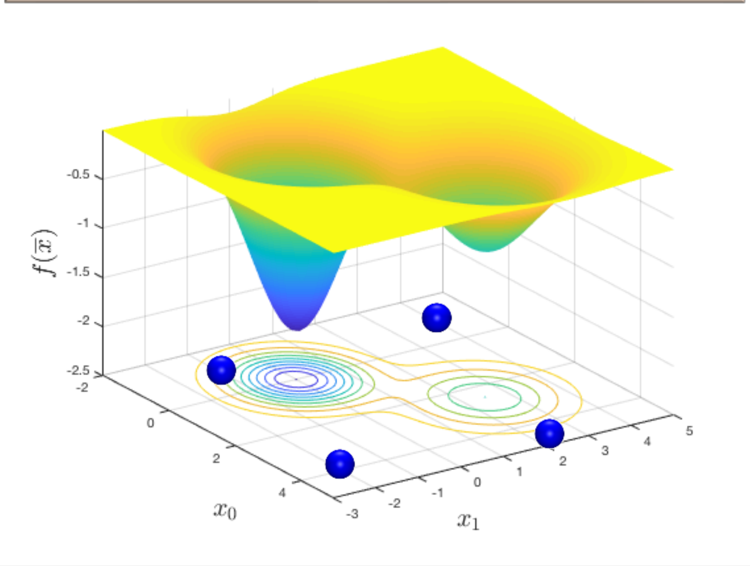
- Genome: position vector
- Genome fitness: Fitness function value at agent location

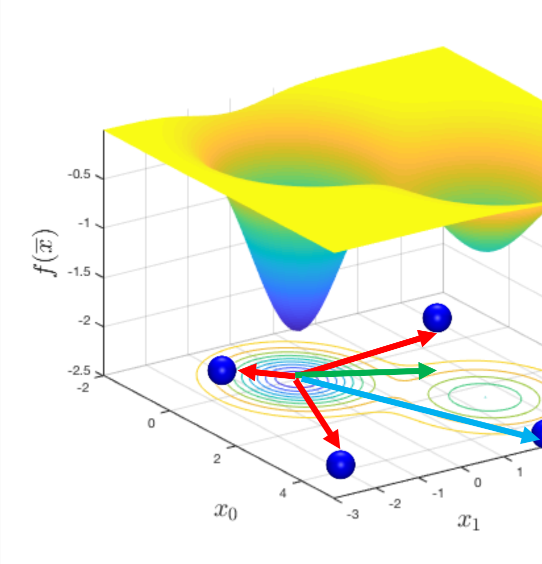
Crossover & Mutation

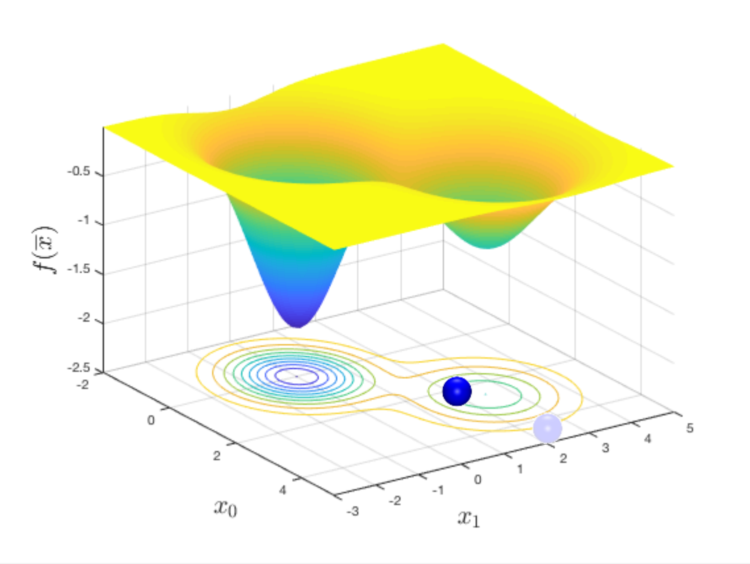
- Linear combination of three random genomes
- Crossover with current particle genome

Selection:

- Select genomes with better fitness

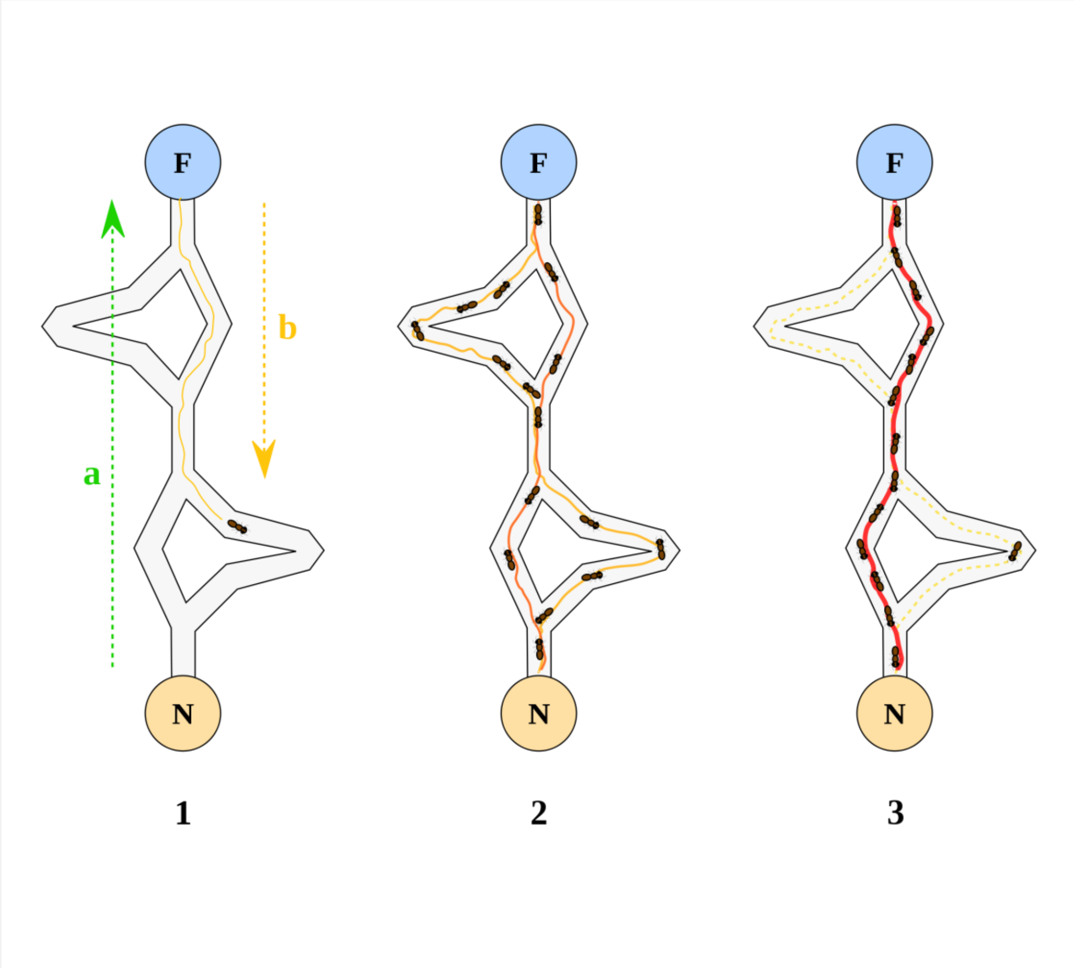






ANT COLONY OPTIMIZATION

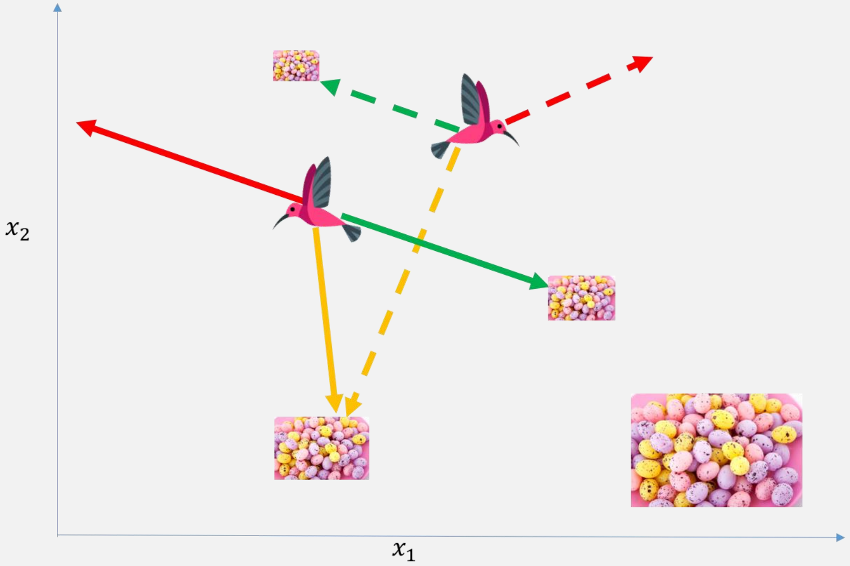
- Based on the foraging behavior of ants
- Each ant leaves a "pheromone" trail
- More pheromone attracts more ants
- Better suited to discrete optimization (like GA)
    - Shortest path problems



http://www.sciencedirect.com/science/article/pii/S0142061515005840

PARTICLE SWARM OPTIMIZATION

Kinematics and Dynamics – Basic PSO



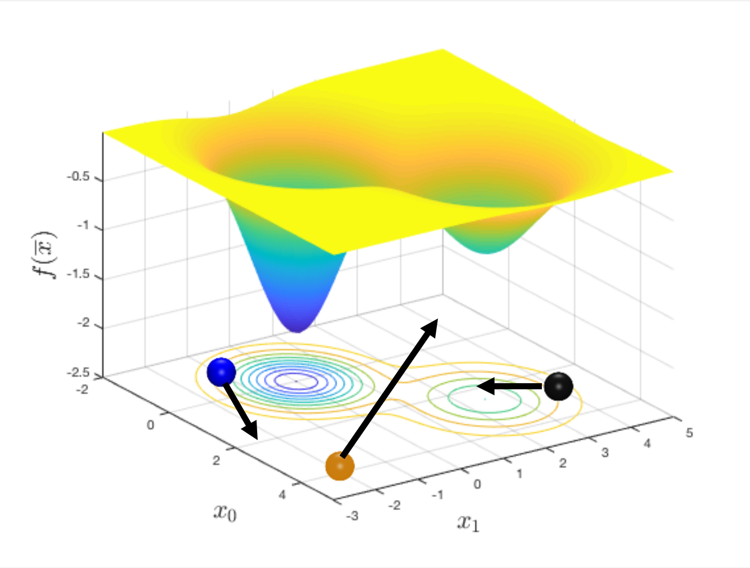
PARTICLE SWARM OPTIMIZATION

- A swarm intelligence method inspired by the flocking behavior of birds
- Search for the biggest food source: Each bird moves under random attraction towards the best food sources that it and the swarm have found but also continues exploration

PARTICLE SWARM OPTIMIZATION

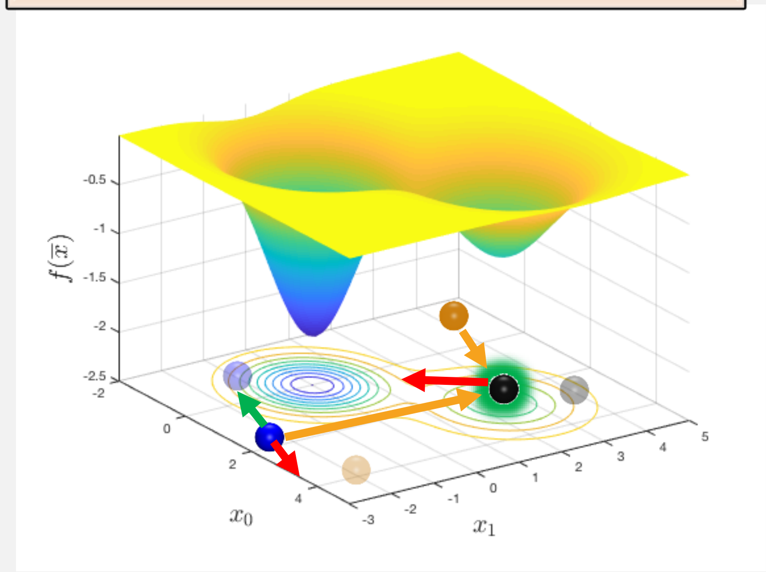
Initialization

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



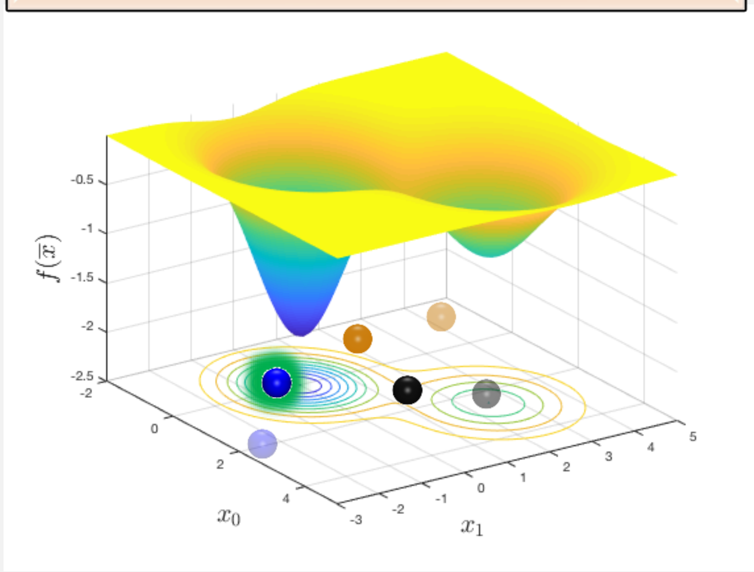
Velocity update

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



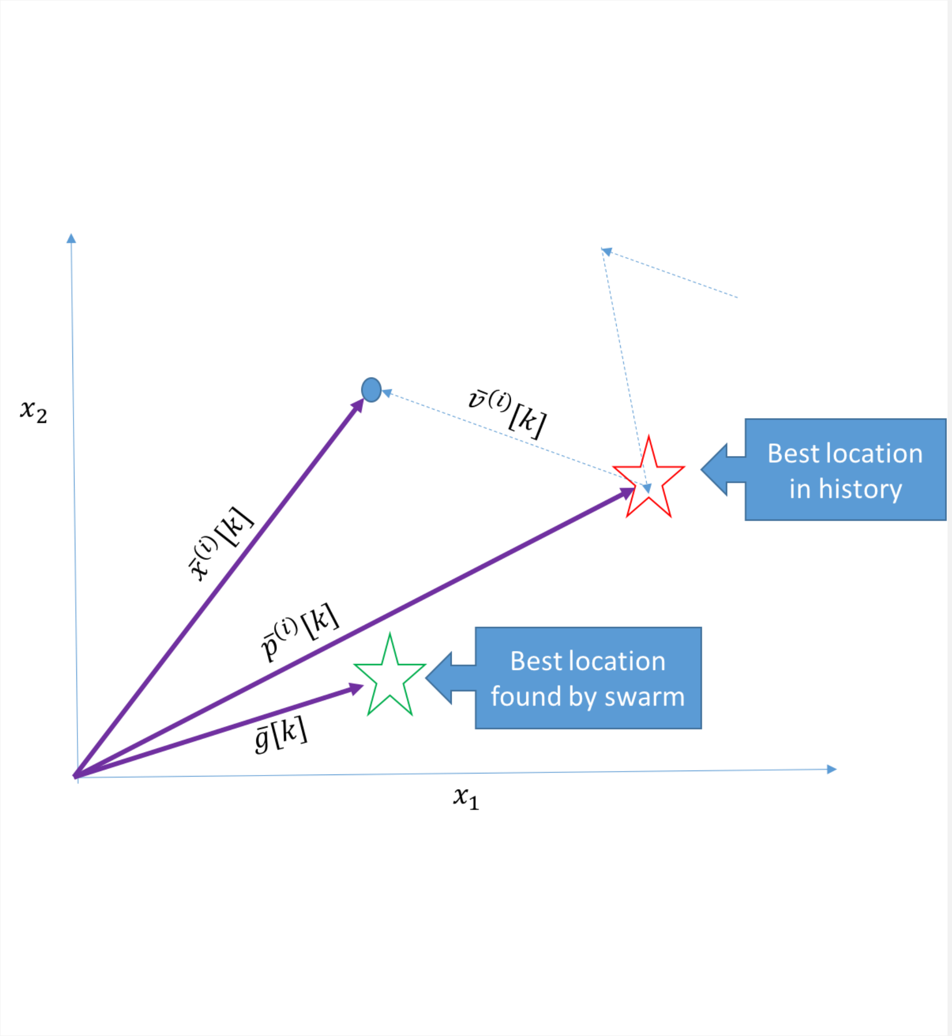
Position update

- Particles move to new positions



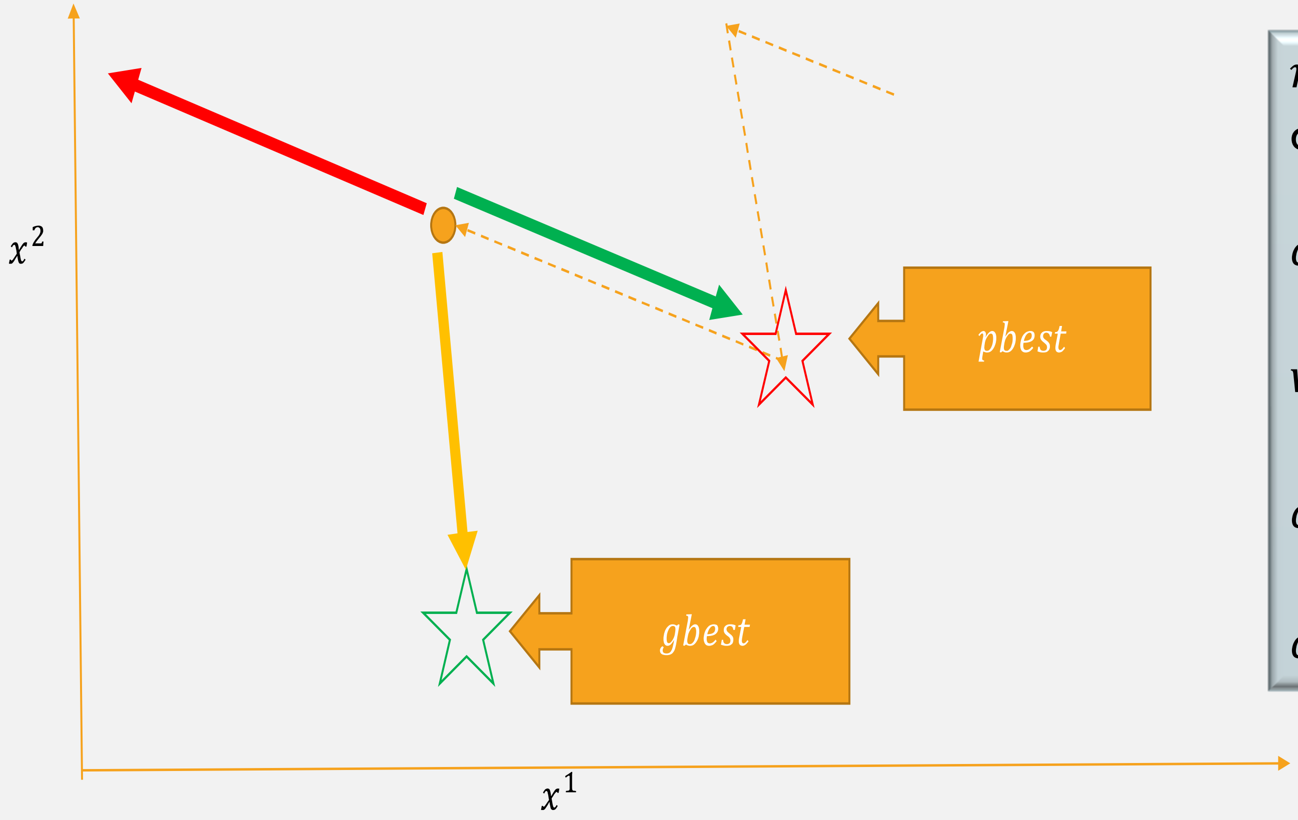
PSO TERMINOLOGY

Term	Definition
Particles	Locations in search space
$\bar{x}^{(i)}[k]$	• Position of i^{th} particle in k^{th} iteration • $\bar{x}^{(i)}[k] = (x_0^{(i)}[k], x_1^{(i)}[k], \ldots, x_D^{(i)}[k])$
$\bar{v}^{(i)}[k]$	• Velocity of i^{th} particle in k^{th} iteration • $\bar{v}^{(i)}[k] = (v_0^{(i)}[k], v_1^{(i)}[k], \ldots, v_D^{(i)}[k])$
$pbest$ ($\bar{p}^{(i)}[k]$)	Personal best: Best location found by the i^{th} particle over iterations 1 through k
$gbest$ ($\bar{g}[k]$)	Global best: Best location found among all particles over iterations 1 through k
v_{max}	Maximum velocity "Velocity Clamping": $v_j^{(i)}[k] \in [-v_{max}, v_{max}]$



VELOCITY UPDATE

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



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

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

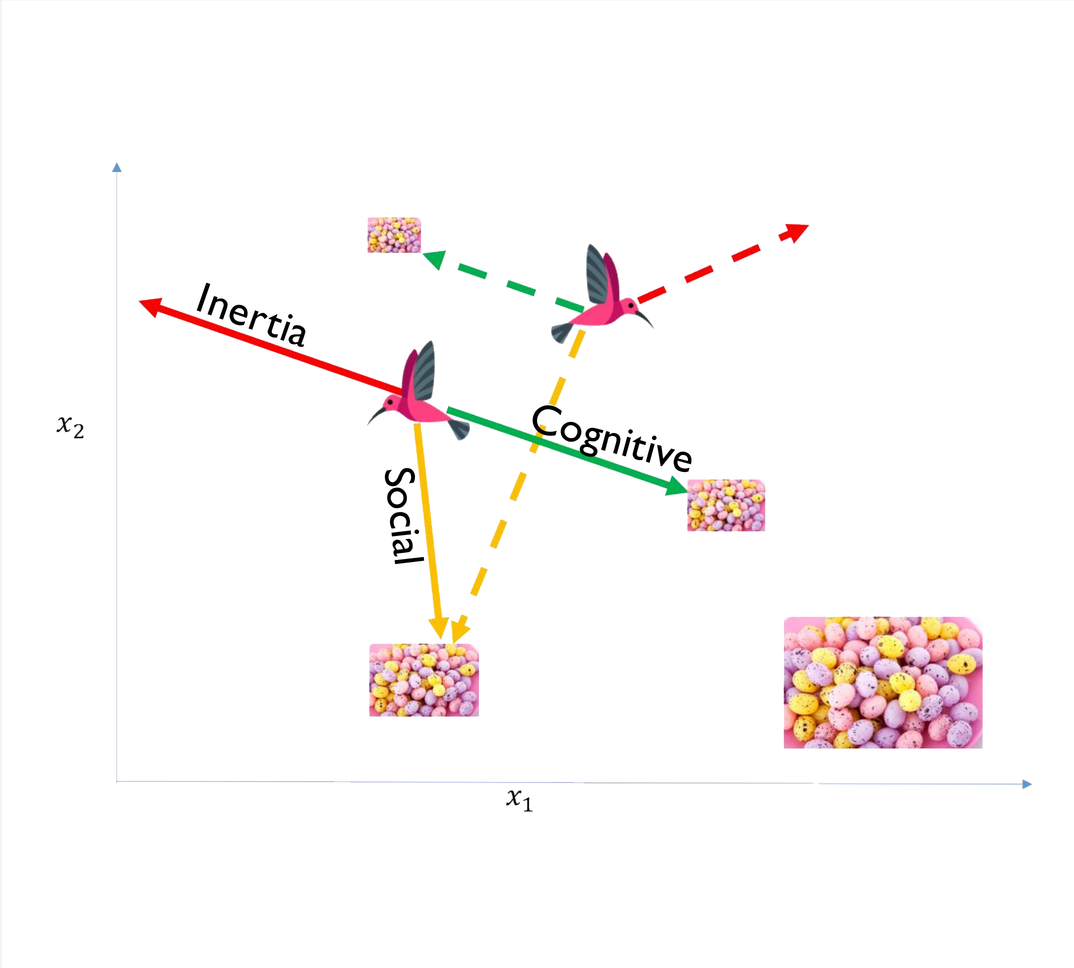
w : "inertia" $\rightarrow w\, v_j^{(i)}[k]$:"**Inertia Term**"

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

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

INTERPRETATION

- Inertia term: promotes exploration
  - $w < 1 \Rightarrow v[k+1] = wv[k] < v[k]$
  - Avoids "particle explosion"
- Social and cognitive terms: promote exploitation
  - Randomization in these terms promotes exploration



PSO DYNAMICAL EQUATIONS

Velocity update

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

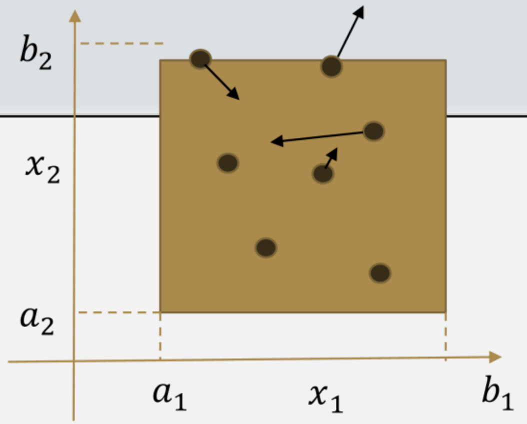
Position update

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

INITIALIZATION

Initialization

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



Initial velocity (variations)

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

TERMINATION (VARIATIONS)

Stop when maximum number of iterations reached

Stop when an acceptable solution has been found

- If x^* is known (e.g., benchmark functions), stop when $|f(x_k) - f(x^*)| < \epsilon$
- Stop if the average change in particle positions is small
- Stop if the average velocity over a number of iterations is close to zero
- Stop if there is no significant improvement in the fitness value over a number of iterations

Stop when the normalized swarm radius is close to zero

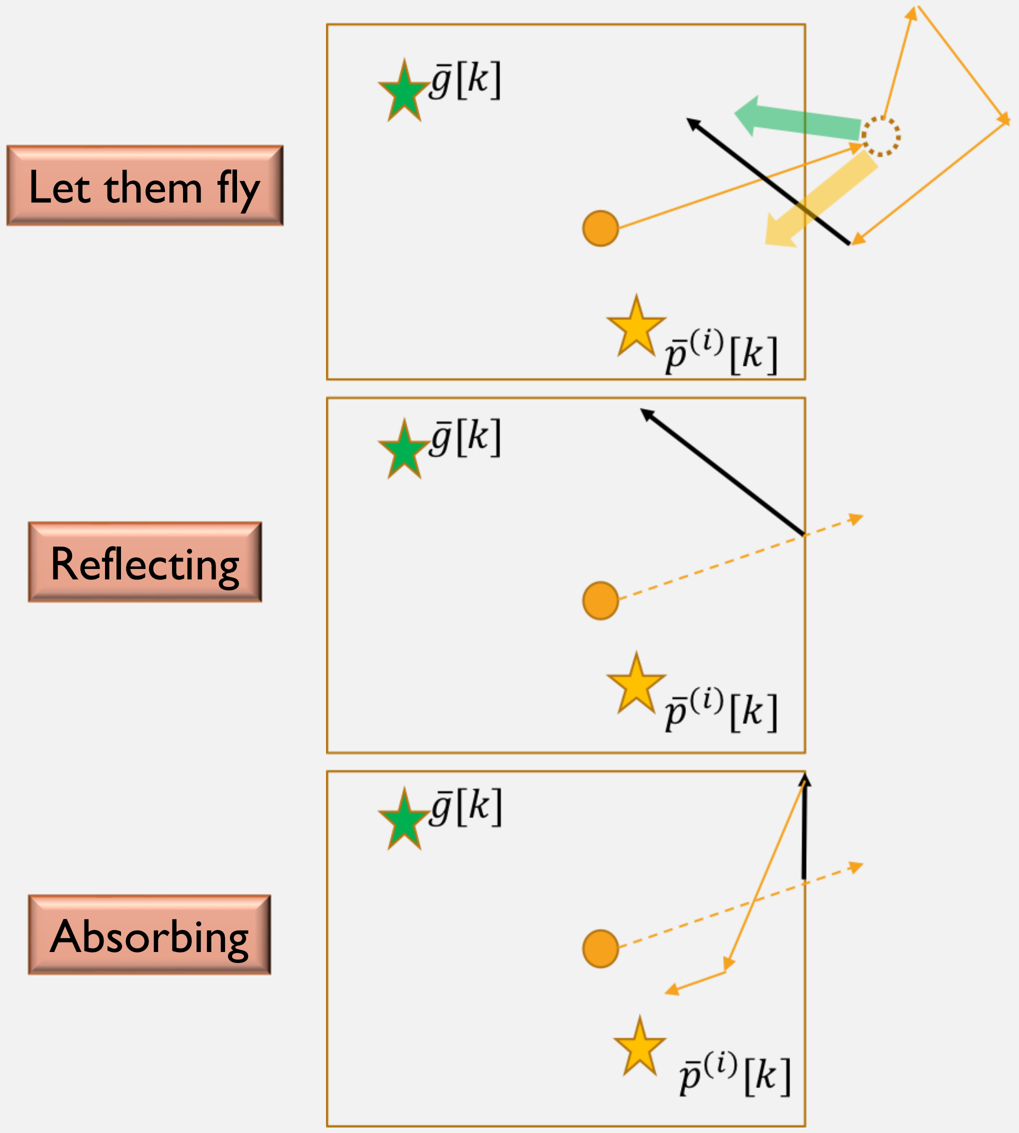
- $R_{norm} = \frac{R_{max}}{\text{initial } R_{max}}; R_{max} = \max_i \left\| \bar{x}^{(i)} [k] - \bar{g} [k] \right\|$

Wang, Mohanty, Physical Review D, 2010

- Stop when the best particle does not move out of a small box over a specified 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

Not a comprehensive review!

VELOCITY CONSTRICTION

- Velocity constriction is another way besides velocity clampling and inertia to contain particle explosion

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

- K is called the constriction factor

$$K = \frac{2}{\left|2 - c - \sqrt{c^2 - 4c}\right|};$$
$$c = c_1 + c_2 > 4$$

- Standard choice for K is 0.729 corresponding to $c = 4.01$
- Normally, $c_1 = c_2 \Rightarrow 2.05$
- Even without velocity constriction, $c_1 = c_2 \simeq 2$ is widely adopted in the literature

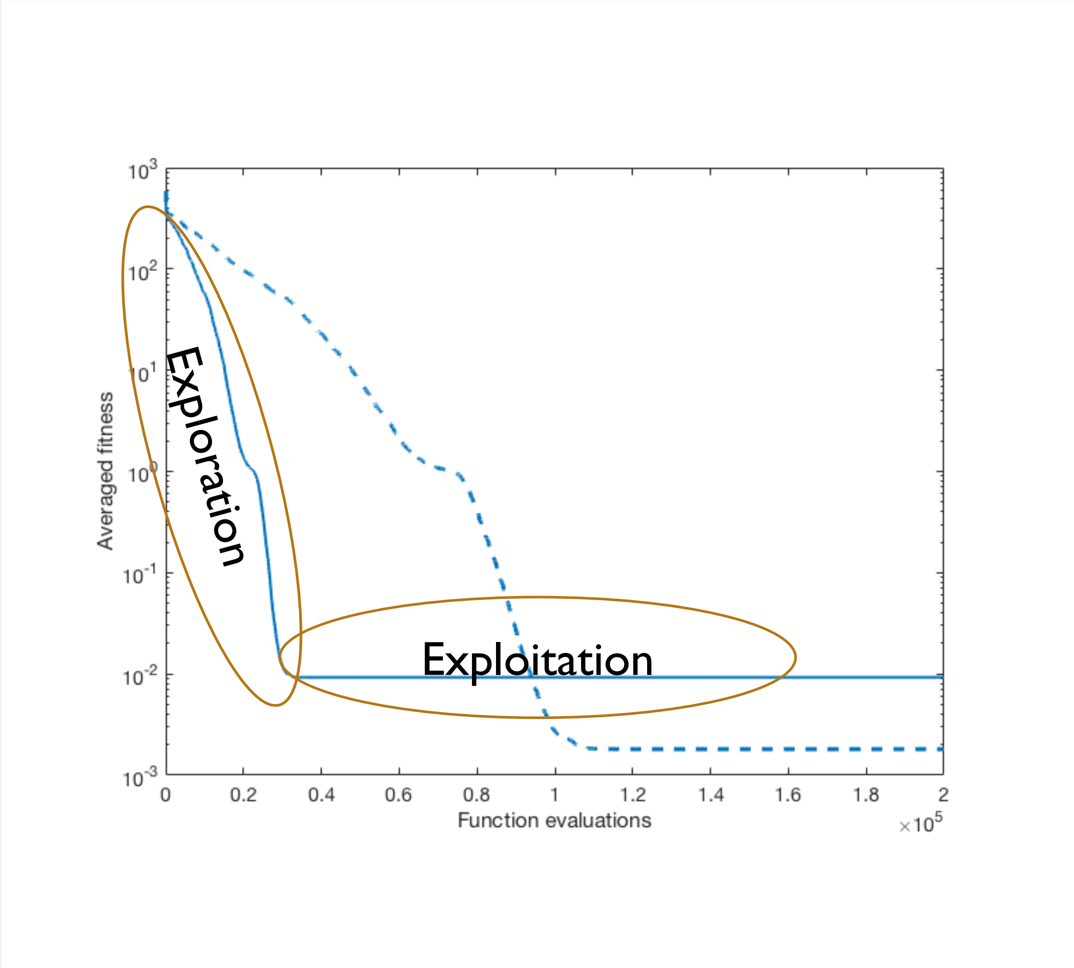
INERTIA DECAY

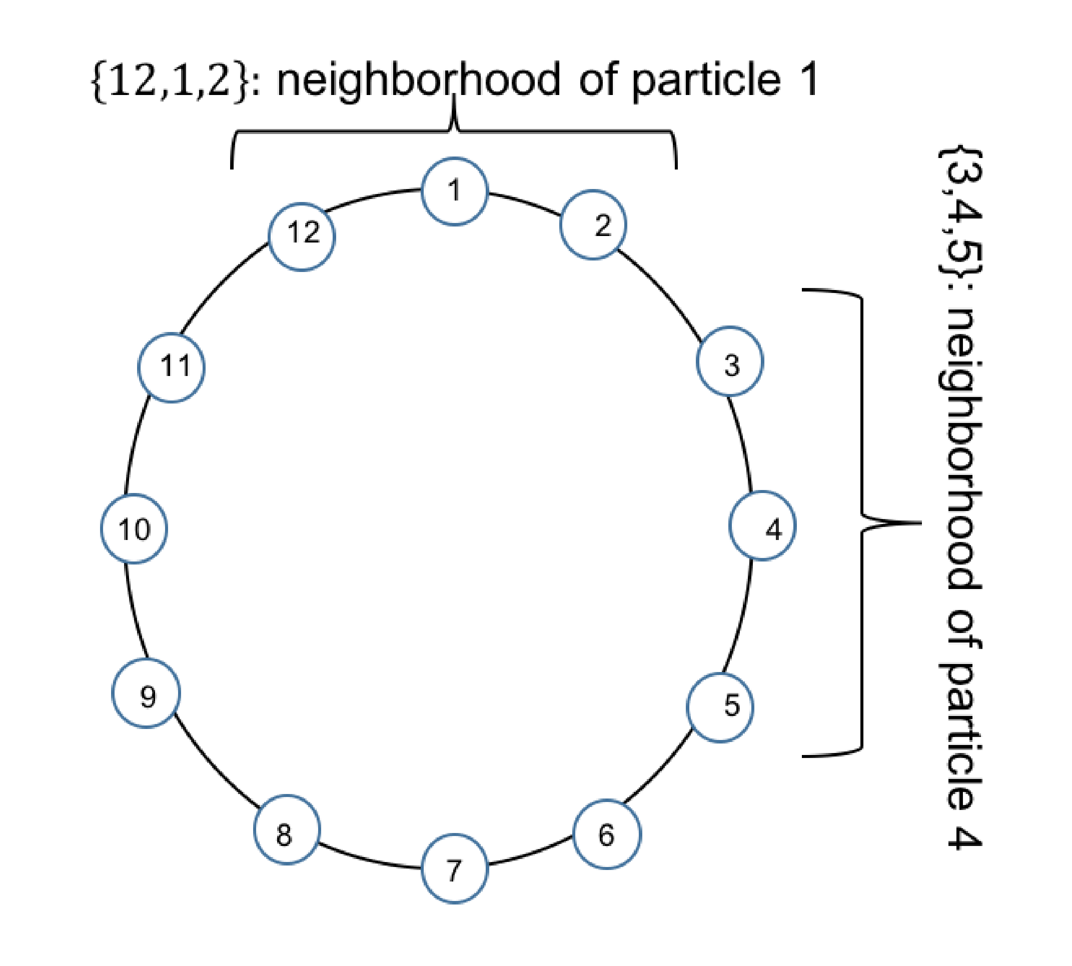
- For termination after a fixed number, N_{iter} , of iterations

<u>Linear decay</u>

$$w \to w[k] = 0.9 - 0.5 \frac{k-1}{N_{iter} - 1}$$

- Transition from exploration to exploitation behavior
- Other laws of inertia decay have been proposed





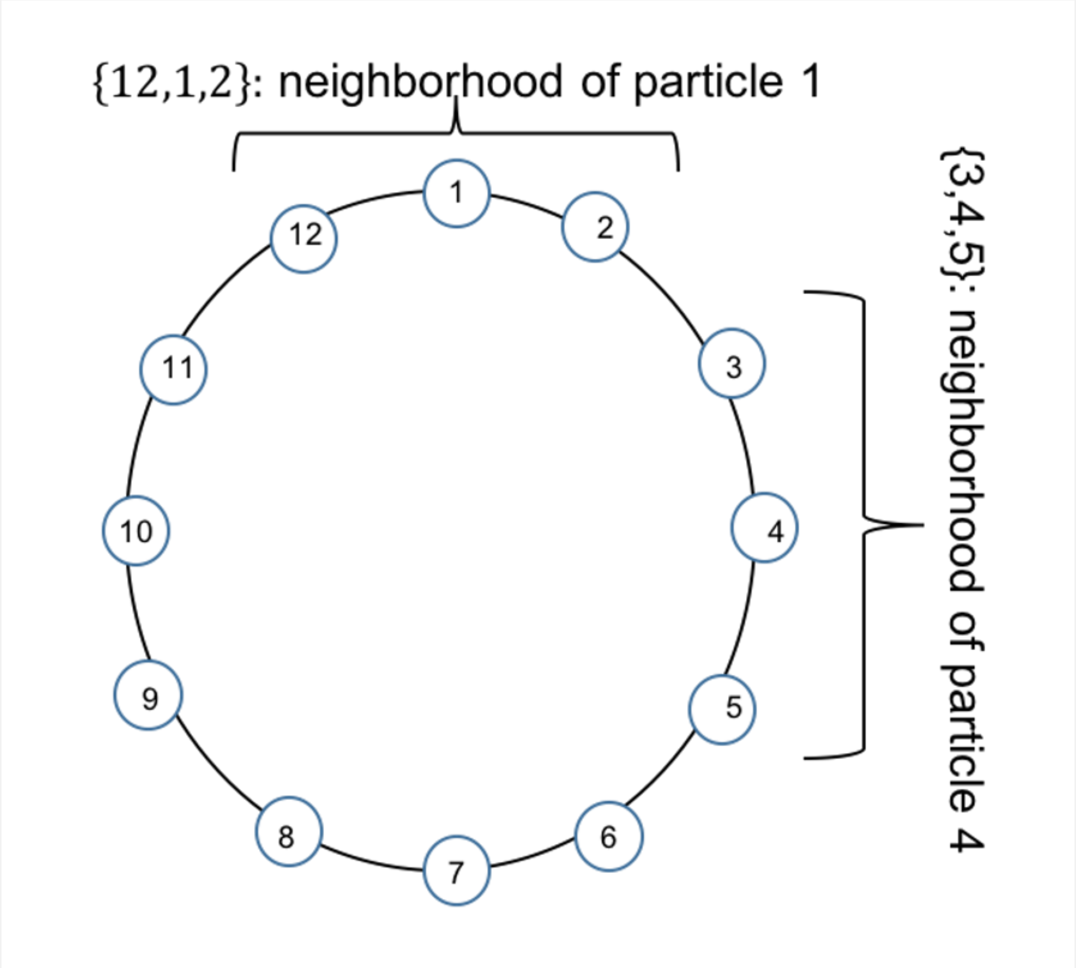
COMMUNICATION TOPOLOGY

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

<u>Local best PSO</u>

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

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



$lbest$ PSO

<u>Local best PSO</u>

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

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

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

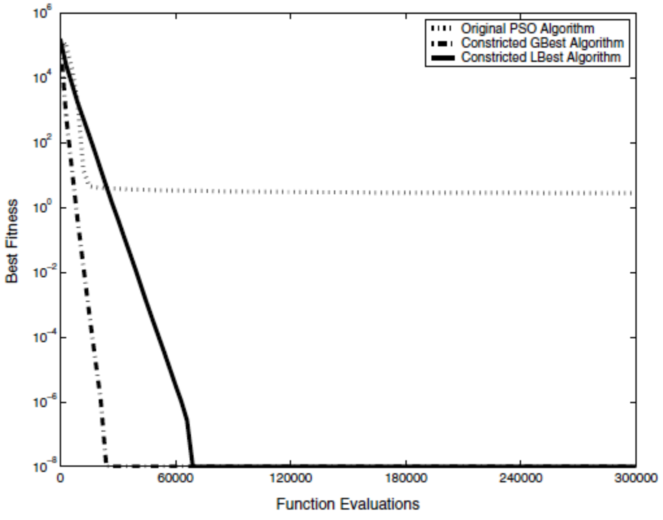
$$(2, 3, 4), (3, 4, 5), \ldots$$

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

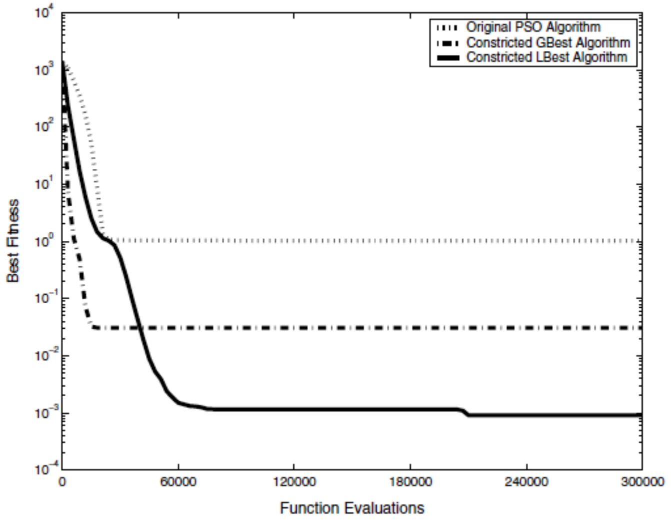
LBEST VS. GBEST PSO

- Tends to become better than gbest PSO as:
  - the dimensionality increases and/or
  - fitness function becomes more rugged
- Penalty: More fitness function evaluations

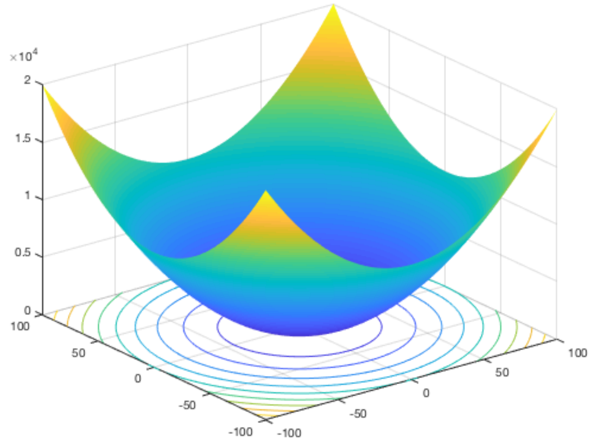
From Bratton & Kennedy, 2007

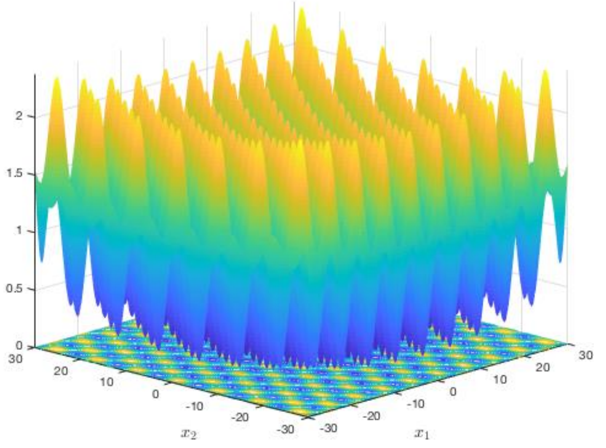


(a) f01 (Sphere/Parabola)



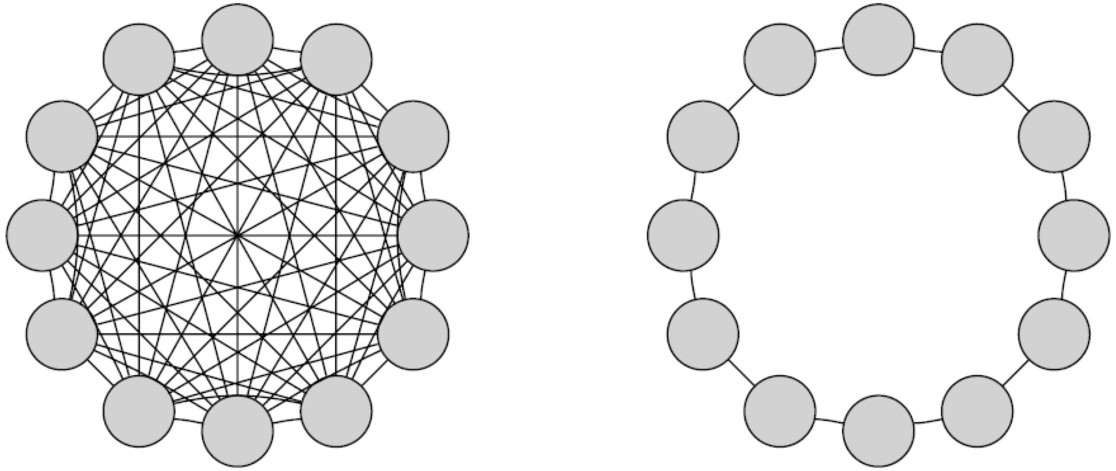
(d) f07 (Generalized Griewank)





TOPOLOGIES

- Gbest
- Ring
- Star
- Wheel
- Pyramid
- Four Clusters
- Von-Neumann
- …

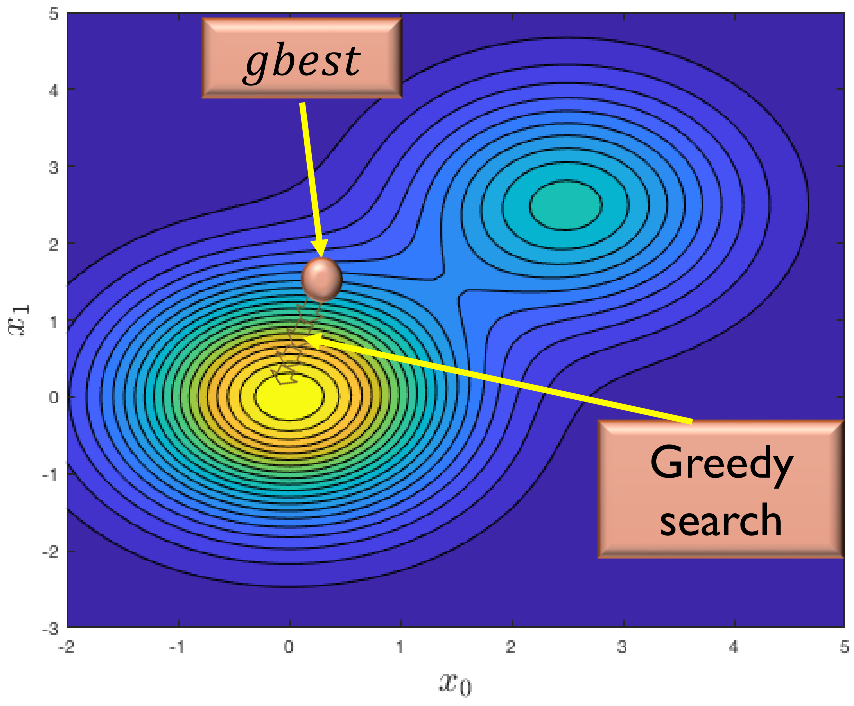


(a) The gbest topology (b) The lbest ring topology

MEMETIC SEARCH

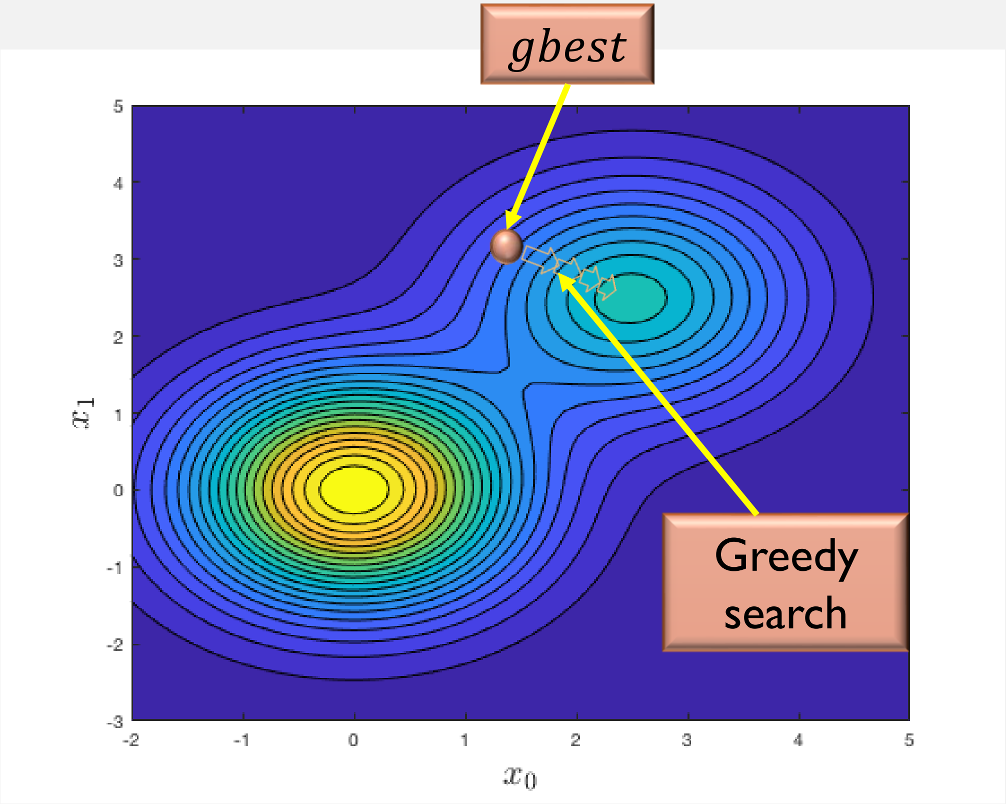
- PSO converges slowly at late stages
- Local optimizers, e.g., steepest descent, can converge to a local minimum much faster

1. Use local search in each iteration to refine $gbest$ or $lbest$, or …
2. Use stochastic local search in a neighborhood of *gbest* or *lbest*



MEMETIC SEARCH

- A danger is finding a good local minimum too early
- $gbest$ stays locked to this local minimum and attracts all the particles
- This shortens the exploration phase and increases the chances of missing the global minimum



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

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_{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

PSO APPLICATIONS

Parametric and non-parametric regression

PARAMETRTIC REGRESSION

- Least-squares fit:

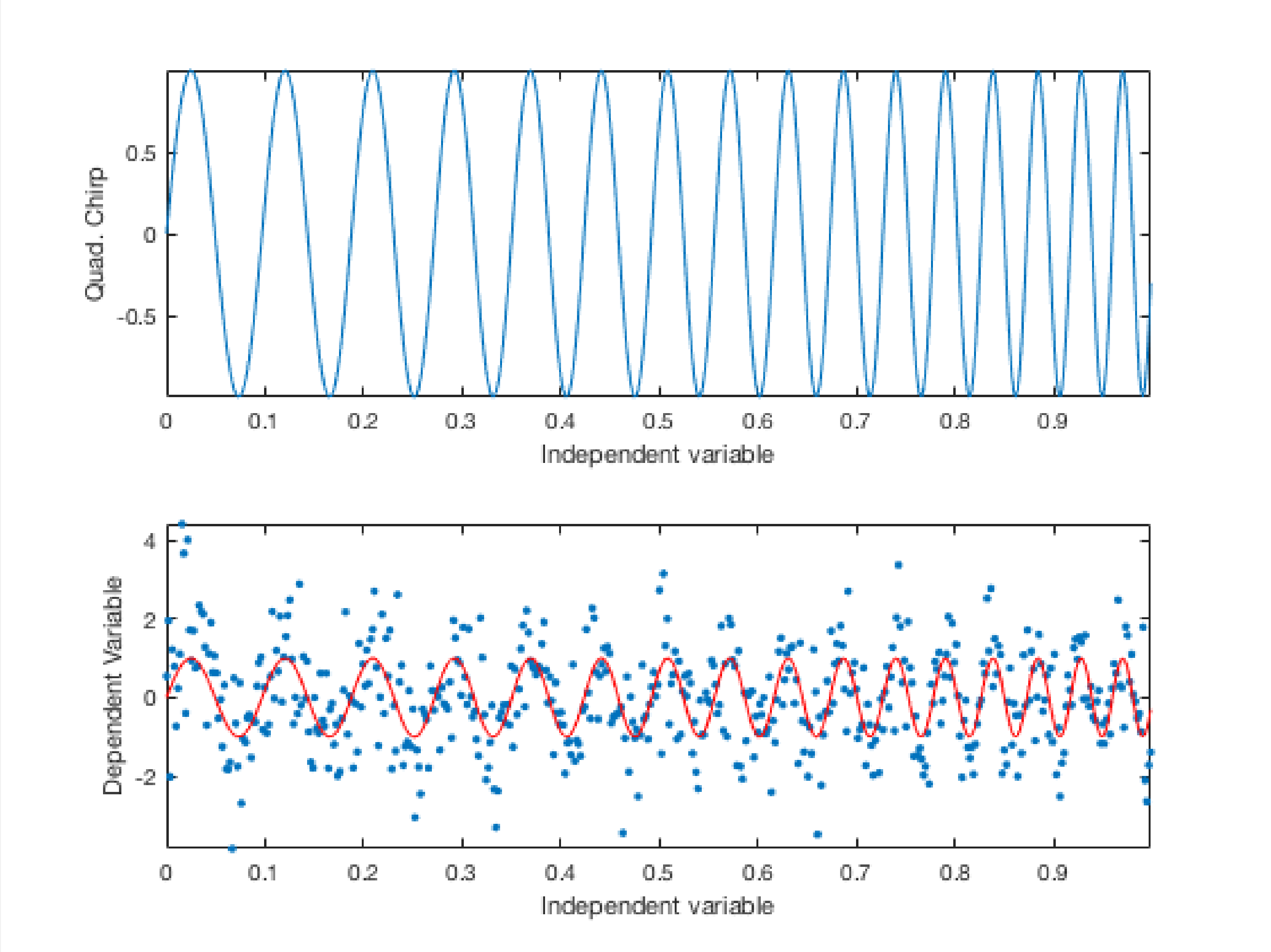
$$\min_{\bar{\theta}} \sum_{i=0}^{N-1} \left(y_i - f(x_i;\bar{\theta})\right)^2$$

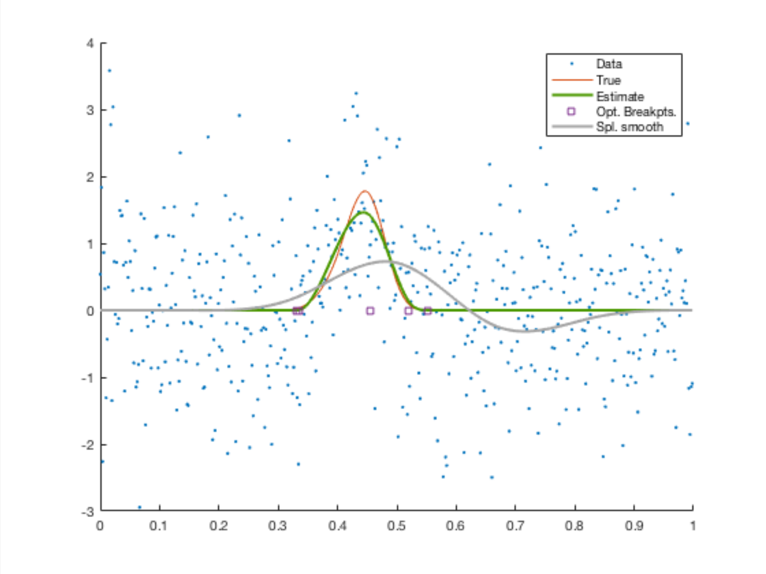
- Non-linear model:

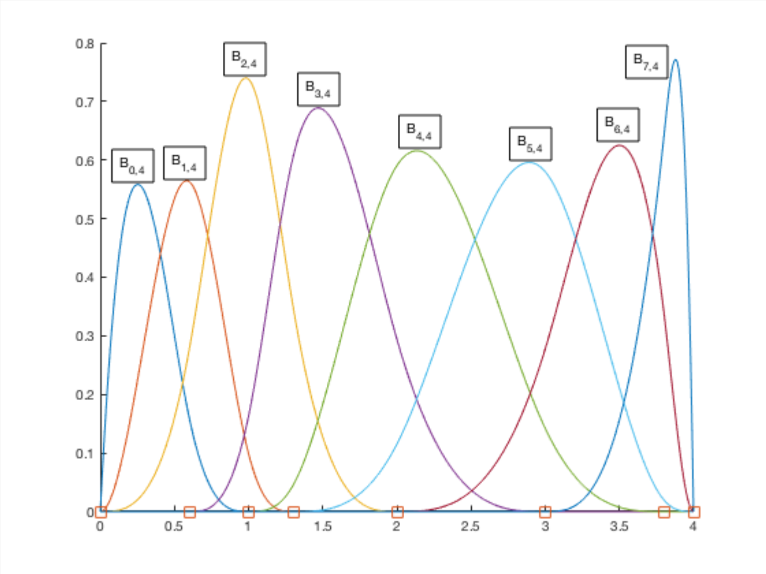
Quadratic chirp (*Lecture 1)

$$f\left(x;\bar{\theta}\right) = A\sin(2\pi\Phi(x))$$

$$\Phi(x) = a_1x + a_2x^2 + a_3x^3$$







NON-PARAMETRIC REGRESSION

- Regression spline (*Lecture 1)

$$f(x;\ \bar{\alpha}, \bar{b}) = \sum_{j=0}^{M-1} \alpha_j B_{j,4}(x; \bar{b})$$

- Least-squares:

$$\min_{\bar{\alpha}, \bar{b}} \sum_{i=0}^{N-1} \left(y_i - f(x_i;\ \bar{\alpha})\right)^2$$

- Fixed number (M) but not fixed locations of breakpoints $(\bar{b})$

STEP 1: ANALYTIC MINIMIZATION

$$\min_{\bar{\theta}} \sum_{i=0}^{N-1} \left(y_i - f(x_i; \bar{\theta})\right)^2$$

Parametric

$$f(x; \bar{\theta}) = A \sin(2\pi\Phi(x))$$

$$\Phi(x) = a_1 x + a_2 x^2 + a_3 x^3$$

- $\min_{(a_1, a_2, a_3)} \left(\min_A \sum_{i=0}^{N-1} (y_i - A \sin(2\pi\Phi(x)))^2 \right)$
- A can be minimized analytically
- PSO handles the optimization over phase parameters $a_i, 1 \leq i \leq 3$

Non-parametric

$$f(x;\ \bar{\alpha}, \bar{b}) = \sum_{j=0}^{M-1} \alpha_j B_{j,4}(x; \bar{b})$$

- $\min_{\bar{b}} \left(\min_{\bar{\alpha}} \sum_{i=0}^{N-1} \left(y_i - \sum_{j=0}^{M-1} \alpha_j B_{j,4}(x; \bar{b})\right)^2 \right)$
- Inner minimization can be done analytically
- PSO handles the optimization over the breakpoints $\bar{b}$

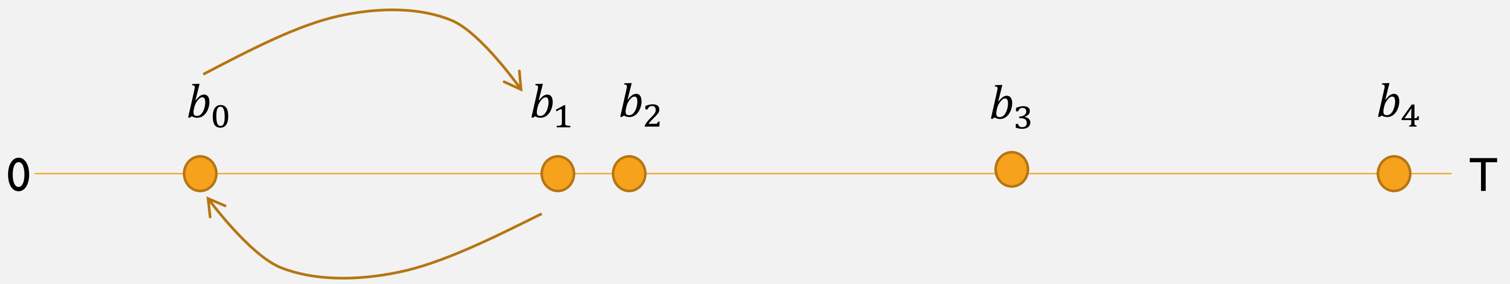
STEP 2: DEGENERACY CONTROL

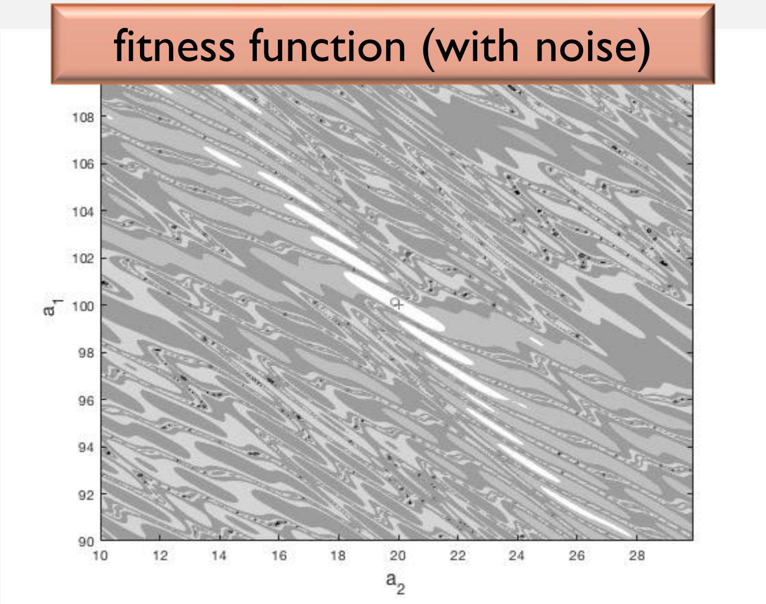
Example: **Unconstrained** optimization over breakpoints

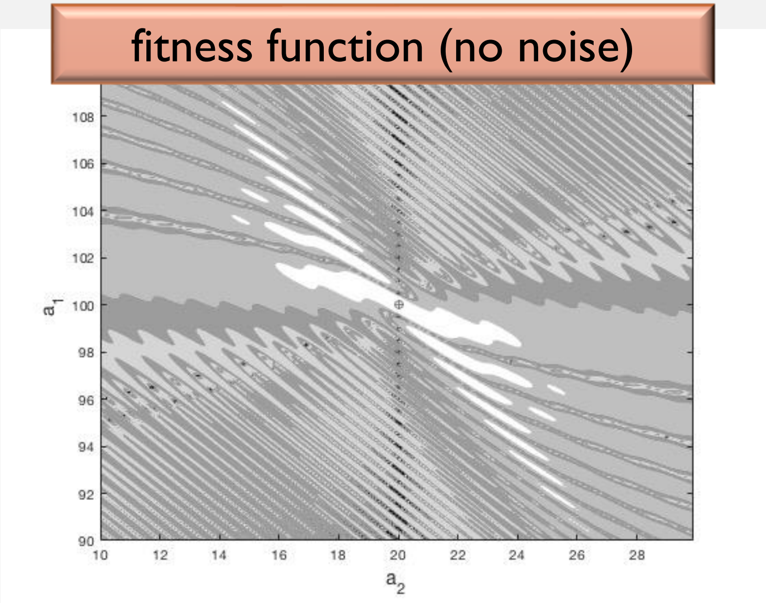
$$\bar{b} = (b_0, b_1, \dots, b_{M-1})$$

Search space: $b_i \in (0, T), \forall i$

- Permutations of $\bar{b}$ correspond to the same spline since the sequence must be ordered before the corresponding spline can be generated
- Degeneracy: Multiple widely-separated points have same fitness values





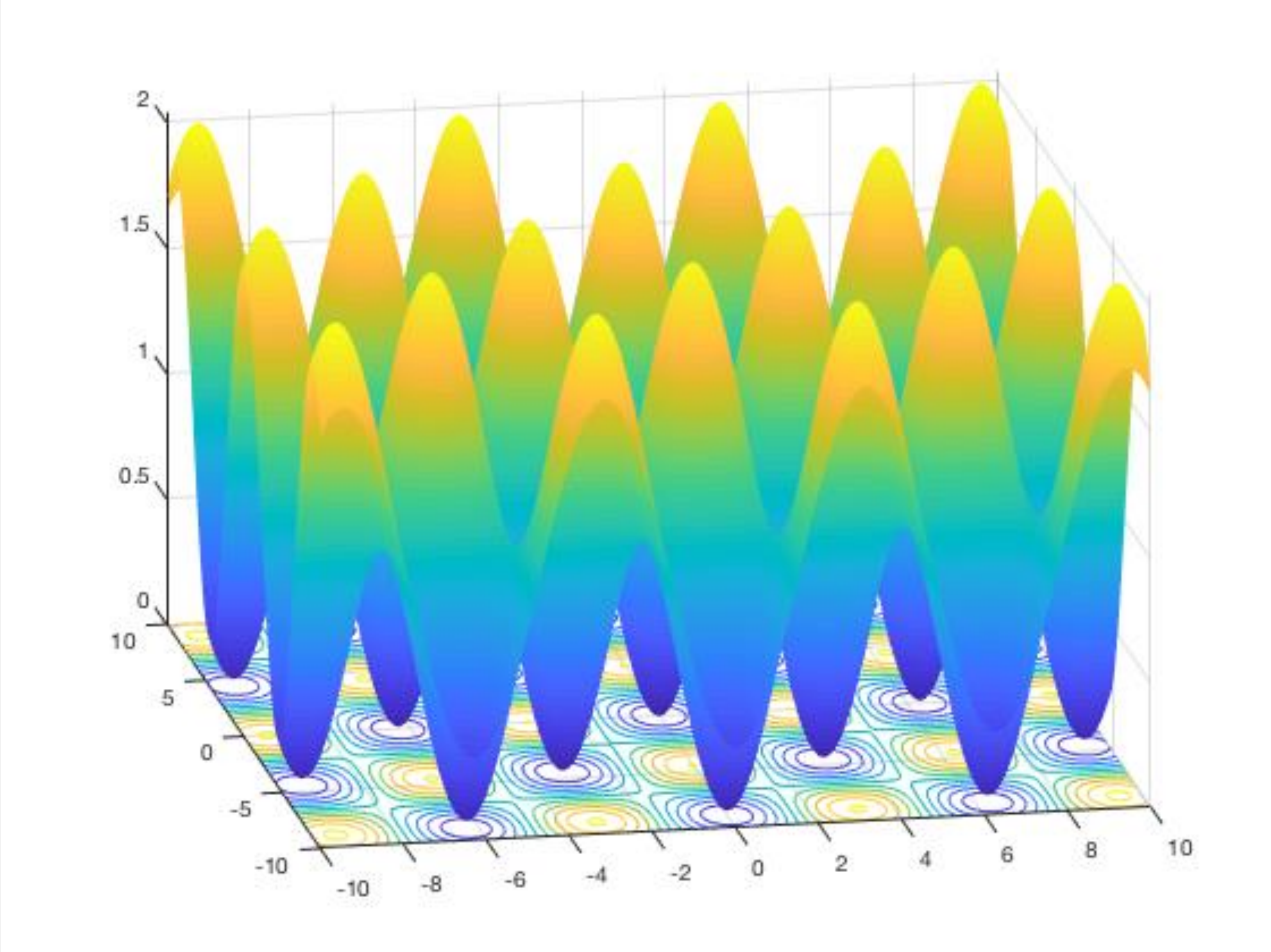


DEGENERACY IN PARAMETRIC REGRESSION

- Quadratic chirp: Multiple local minima in fitness function even in the absence of noise
- Multiple local minima are a hallmark of non-linear regression
- Note: Degeneracy is not restricted to equally deep minima

FITNESS FUNCTION DEGENERACY

- Degeneracy of a fitness function in the absence of noise leads to multiple local minima
- A stochastic optimization method must escape from local minima
- ⇒ Multiple local minima make the search for the global minimum harder

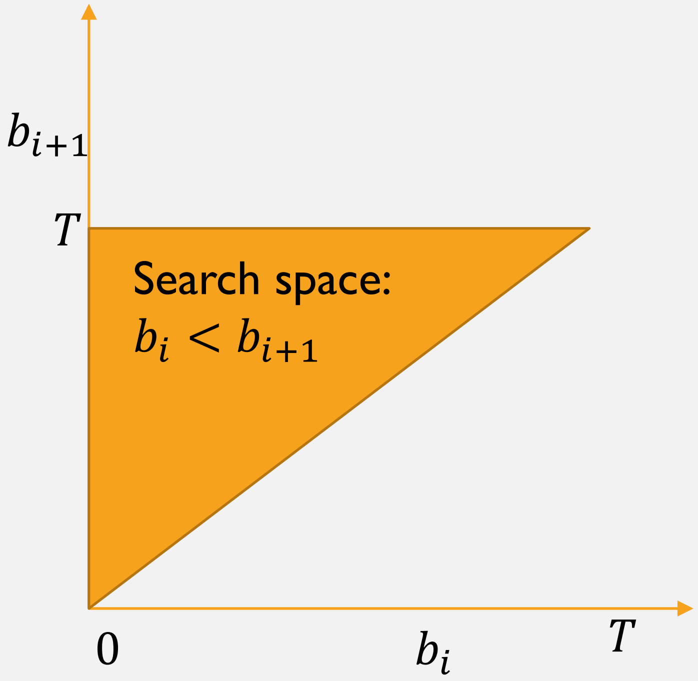


CONSTRAINED SEARCH

- 1^{st} solution to degeneracy in regression spline: Constrained search

$$b_i < b_{i+1}$$

- $\Rightarrow$ Search space shape is a simplex in M dimensions
- PSO does not perform well when the search space is not a hypercube
  - $\Rightarrow$ Excessive leakage of particles from the search space

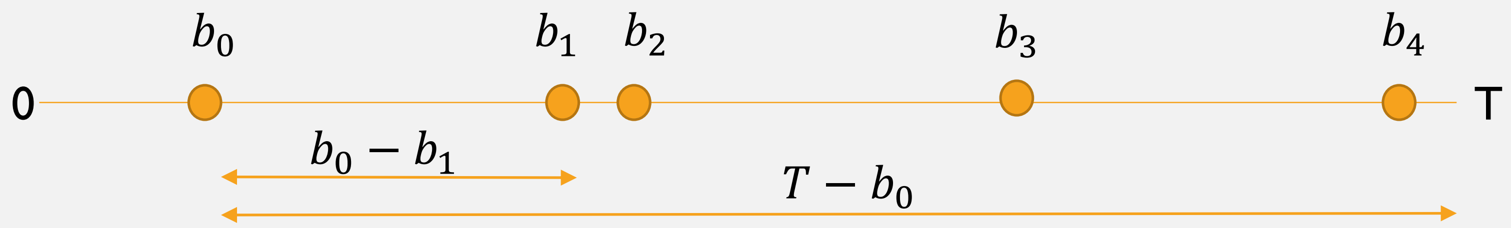


REPARAMETRIZATION

- 2nd solution: Reparametrization

$$\alpha_i = \frac{b_i - b_{i-1}}{T - b_{i-1}} \; ; \; \alpha_0 = \frac{b_0}{T}$$

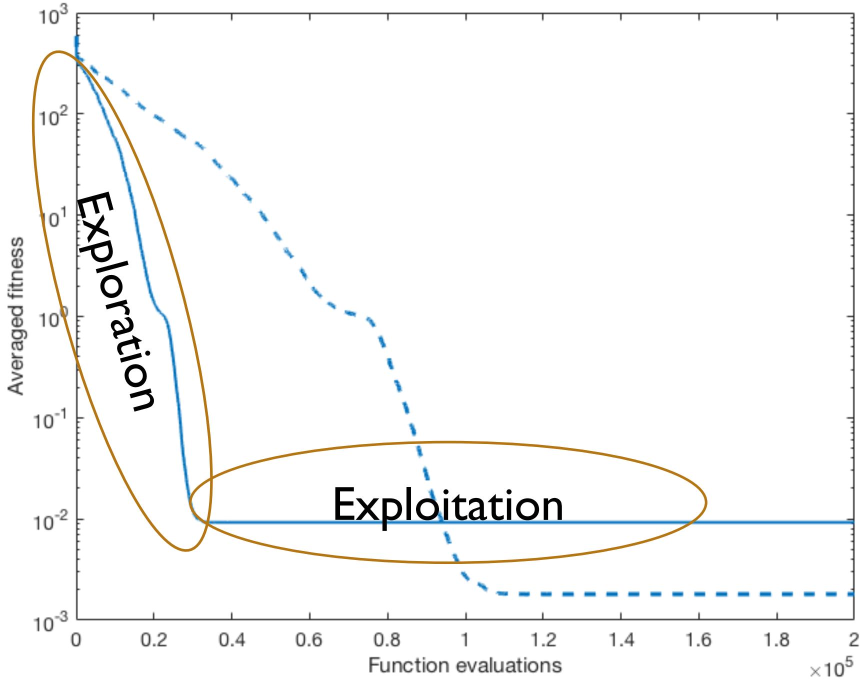
- Guarantees ordered breakpoint sequence while keeping the search space hypercubical: $\alpha_i \in (0,1); \forall i$

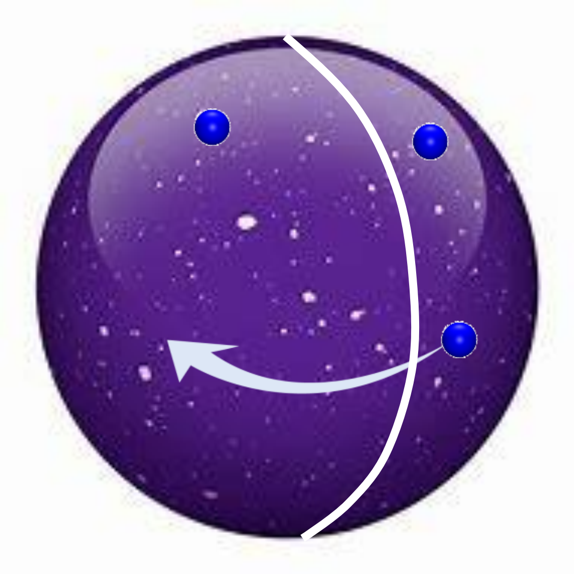


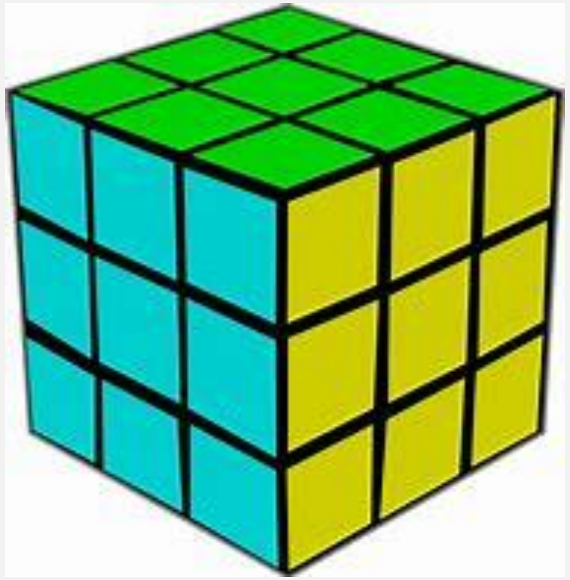
*Further reparametrization (see book): center the uniformly spaced breakpoint sequence

STEP 3: PSO VARIANT

- Exploration-exploitation trade-off
- Examine the extent of degeneracy to get an idea of which PSO variant to use
- Greater ruggedness of fitness function ⇒ Use longer exploration phase
  - Example: Use *lbest* PSO to extend exploration phase
  - *Other options such as slower inertia decay

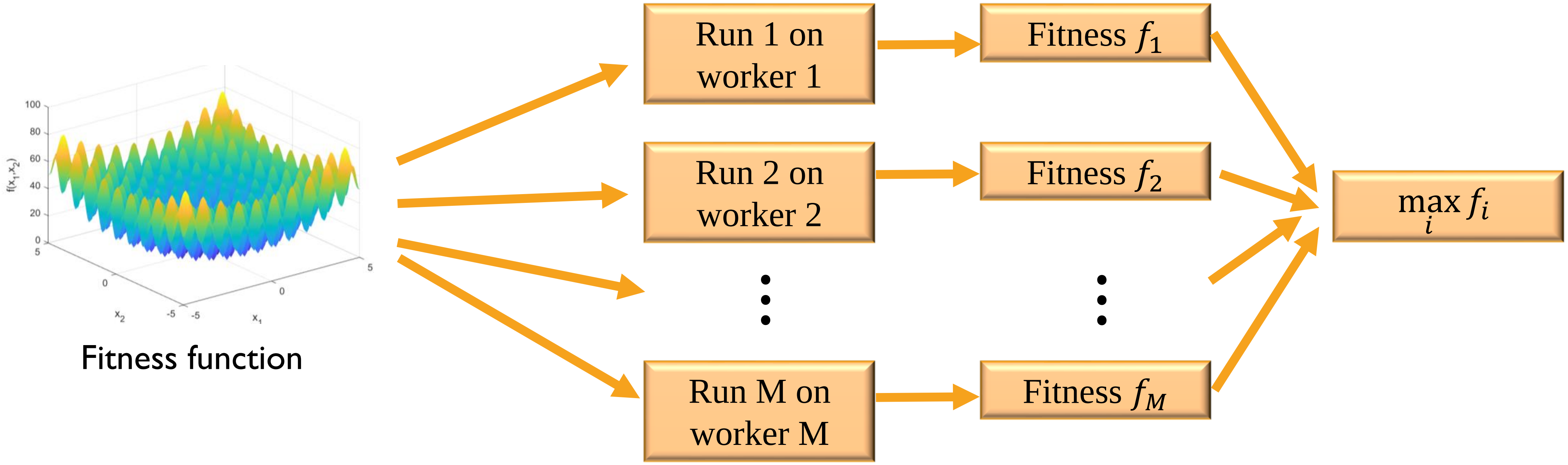






PSO VARIANTS

- If the fitness function has a periodic dependence on a variable, the corresponding boundary condition should be periodic
  - Very helpful in the case of gravitational wave searches because two of the parameters are sky angles
- Consider splitting the search space into smaller domains



STEP4: TUNING

Recommended: Best-of-M-runs (BMR)

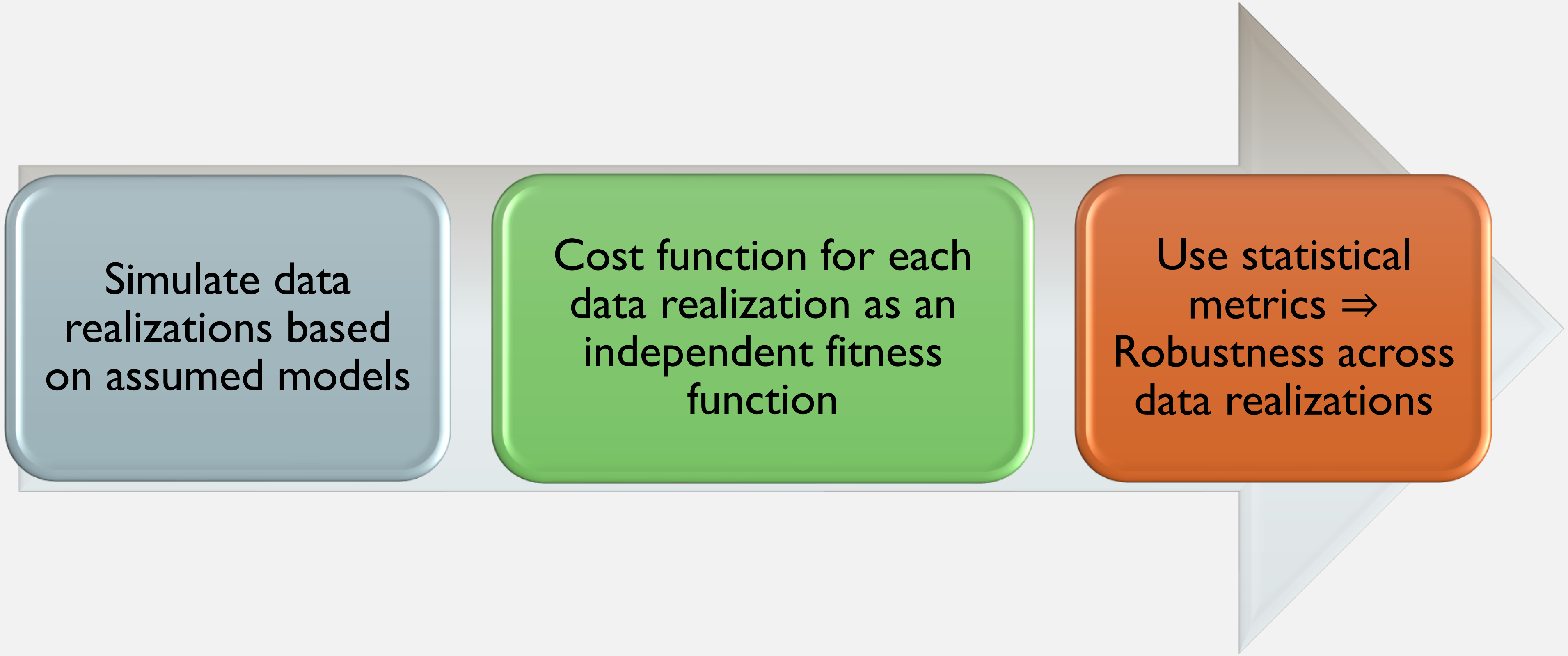
BMR TUNING OF PSO

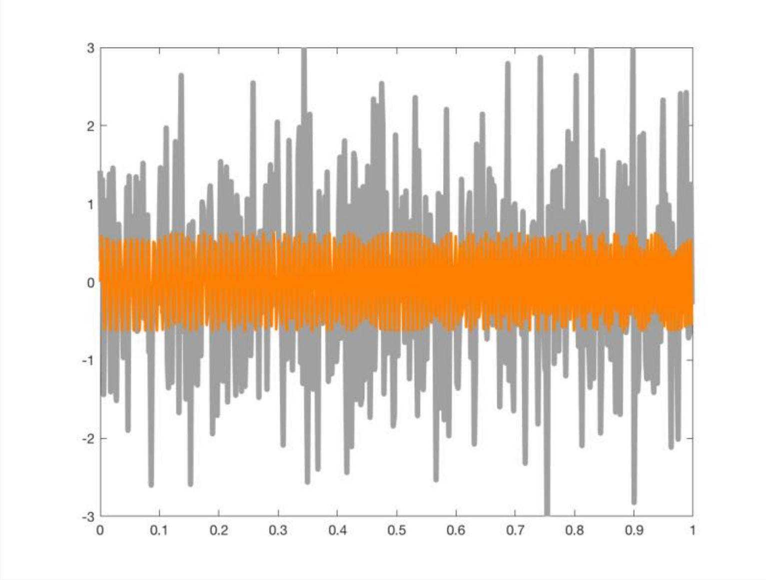
- PSO parameters have robust values and do not need to be changed in most cases
- Two main parameters to tune
  - Number of iterations: N_{iter}
  - Number of independent PSO runs in BMR: N_{runs}

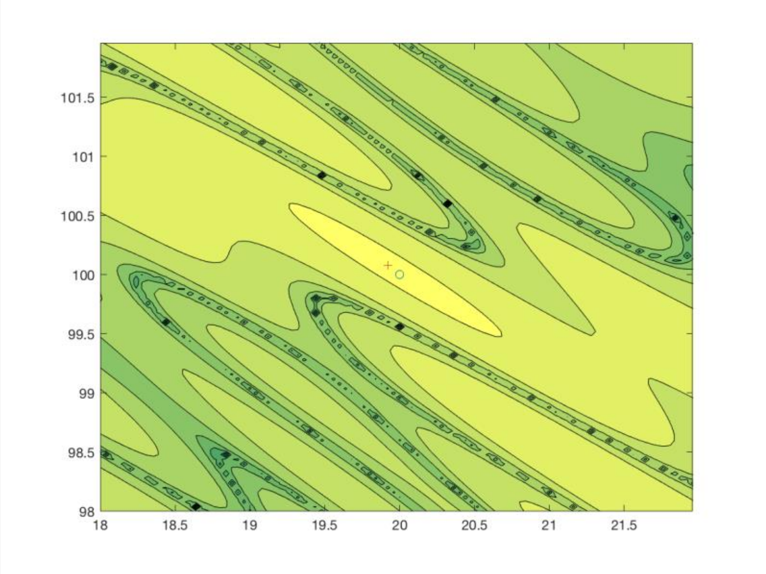
Setting Name	Setting Value
Position initialization	$x_j^{(i)}[0]$ drawn from $U(x;0,1)$
Velocity initialization	$v_j^{(i)}[0]$ drawn from $U(x;0,1) - x_j^{(i)}[0]$
$v_{\rm max}$	0.5
$N_{\rm part}$	40
$c_1 = c_2$	2.0
$w[k]$	Linear decay from 0.9 to 0.4
Boundary condition	Let them fly
Termination condition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

TUNING PSO FOR REGRESSION

TUNING FOR REGRESSION PROBLEMS

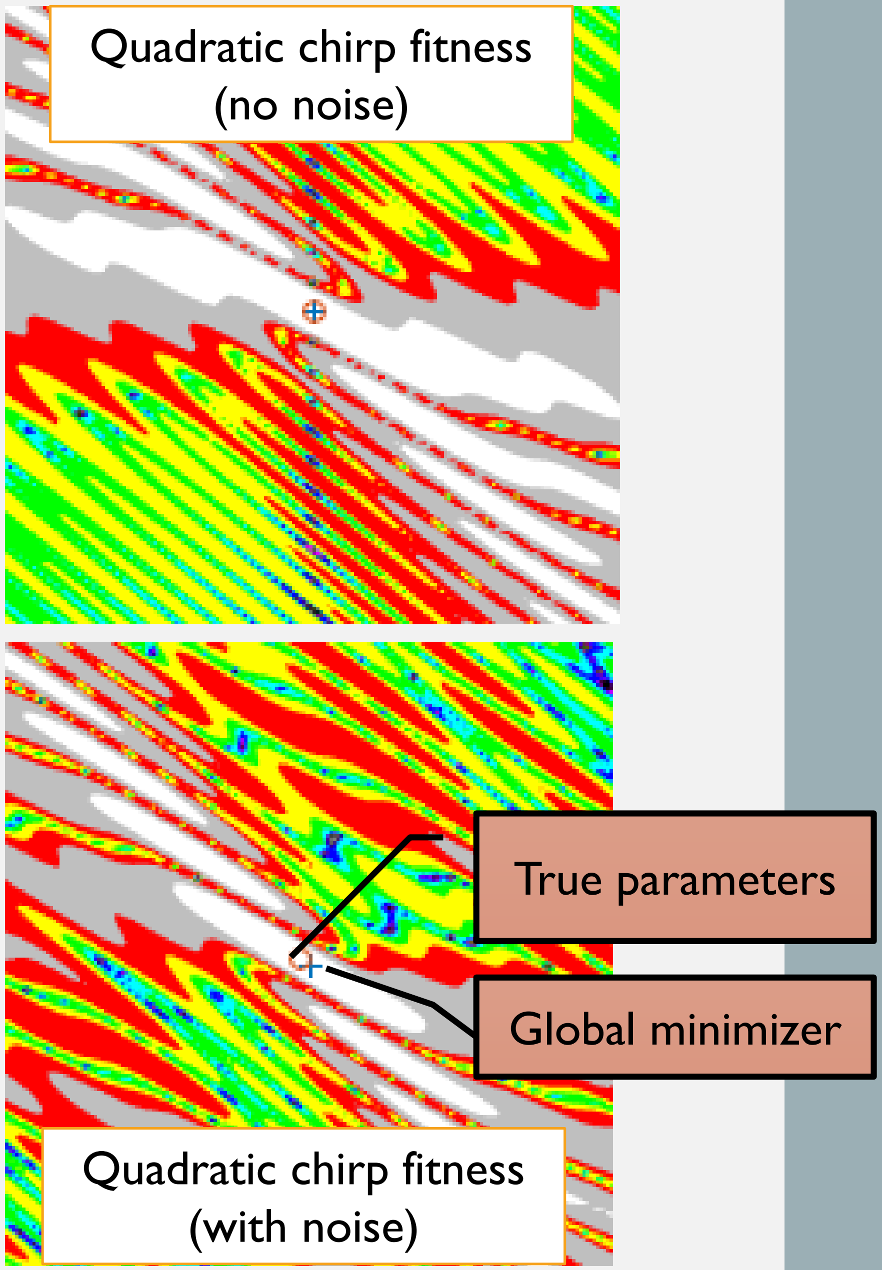






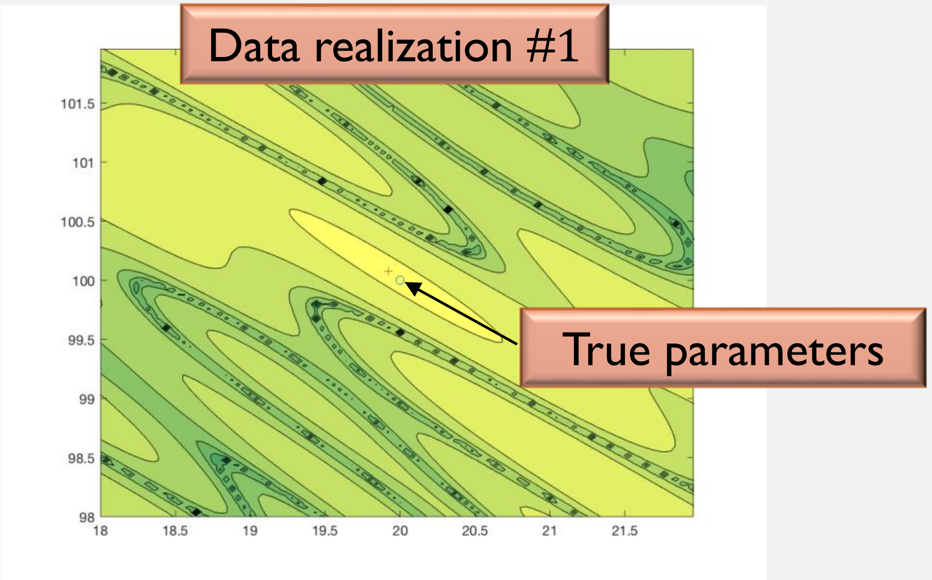
DATA SIMULATION

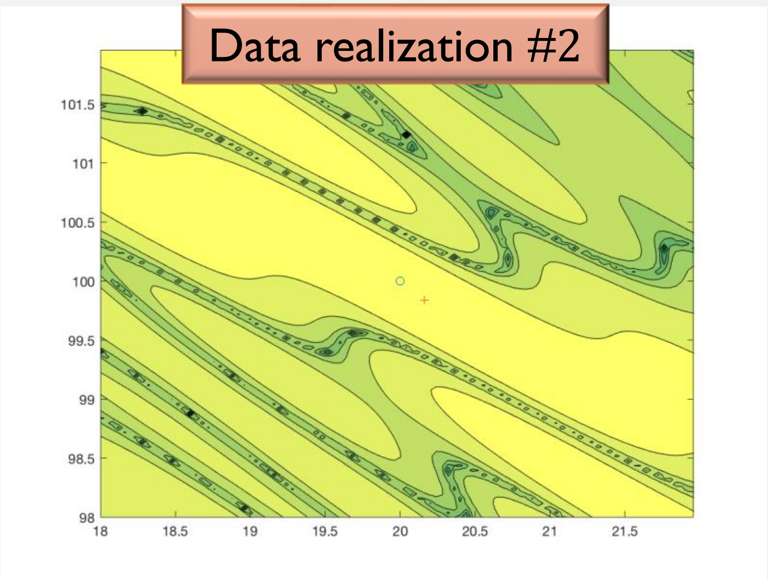
- In the case of the regression examples:
  - Keep the parameters of the true signal (quadratic chirp or spline) fixed
  - Add different noise realizations (pseudo-random numbers)
- Each data realization ⇒ one fitness function realization



STATISTICAL TUNING APPROACH

- For the same true signal parameters, the global minimizer will be different for different data realizations
- The best fitness value will always occur away from the true parameters
  - This is why we get error in parameter estimates in the presence of noise
- This fact can be used to develop a tuning procedure that is well-suited to regression



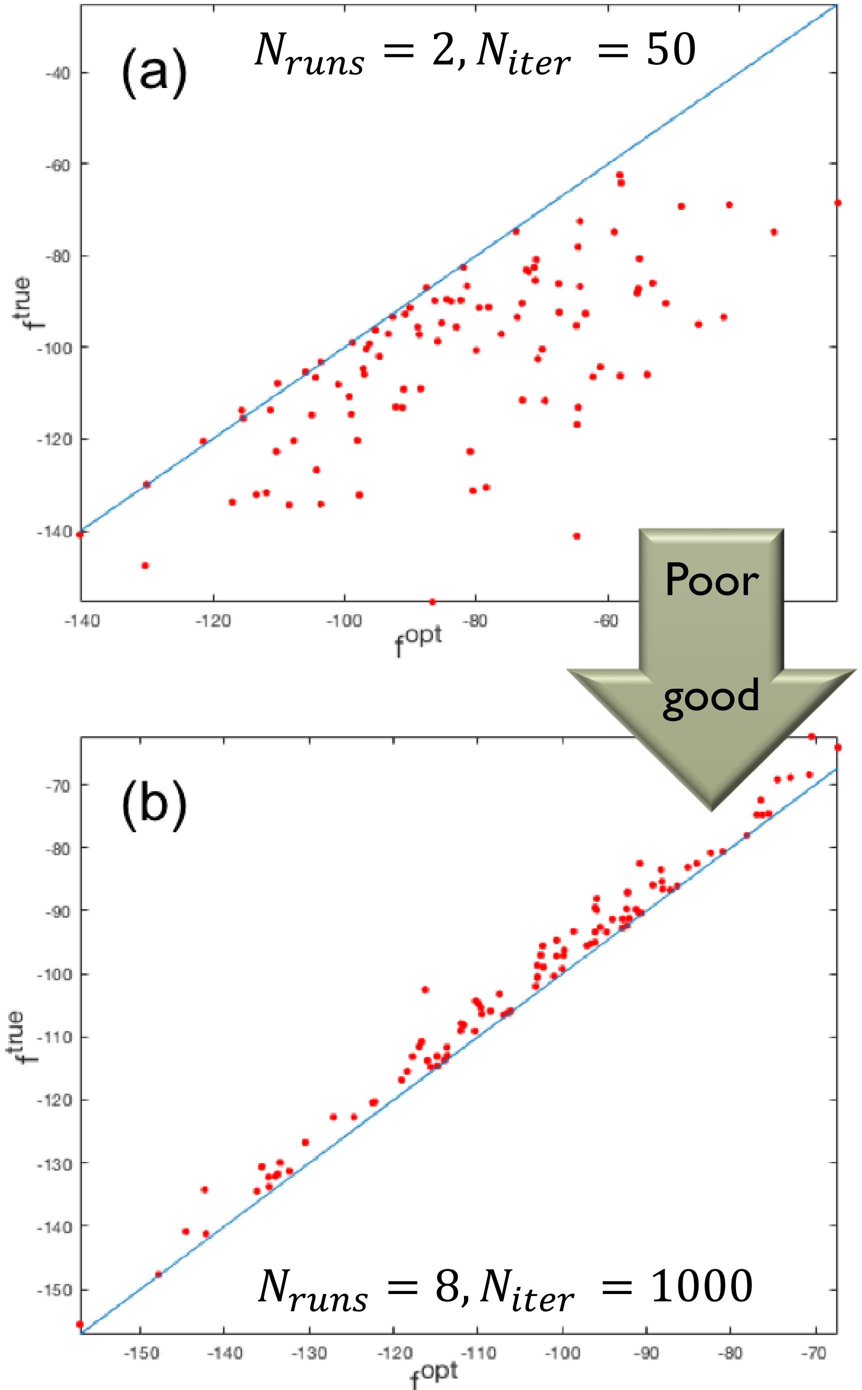


PSO TUNING FOR REGRESSION PROBLEMS

- Key idea: The global minimum must be lower than the fitness at the true parameters

$$f^{opt} < f^{true}$$

- PSO is working well if this condition is satisfied for a sufficiently high fraction of data realizations
- Proposed in:
  - Wang, Mohanty, Physical Review D, 2010
  - Normandin, Mohanty, Weerathunga, Physical Review D, 2018



PARAMETRIC REGRESSION

- The true parameters are known for simulated data
- ⇒Possible to check $f^{opt} < f^{true}$ for each data realization
- Set up a grid of values in
  - N_{iter}: Number of iterations
  - N_{runs}: Number of runs in BMR strategy
- For each combo (N_{iter}, N_{runs}): Get fraction X of N data realizations where this condition is satisfied
- Get all (N_{iter}, N_{runs}) for which X is below some preset value
- Pick the combo in this set with the lowest computational cost

NON-PARAMETRIC REGRESSION

- No explicit parametrization of models ⇒ $f^{opt} < f^{true}$ not easy to check
- The non-parametric fit may never capture every feature of the true signal ⇒ Fitness will typically be worse than fitness for true signal
- Further development of the basic idea needed: $f^{opt} \in f^{true} + [-\epsilon, \epsilon]$?
- ⇒ Seat-of-the-pants approach but not very difficult for PSO due to only two tuning parameters: N_{iter} and N_{runs}

RESULTS

Parametric and non-parametric regression

NON-PARAMETRIC REGRESSION

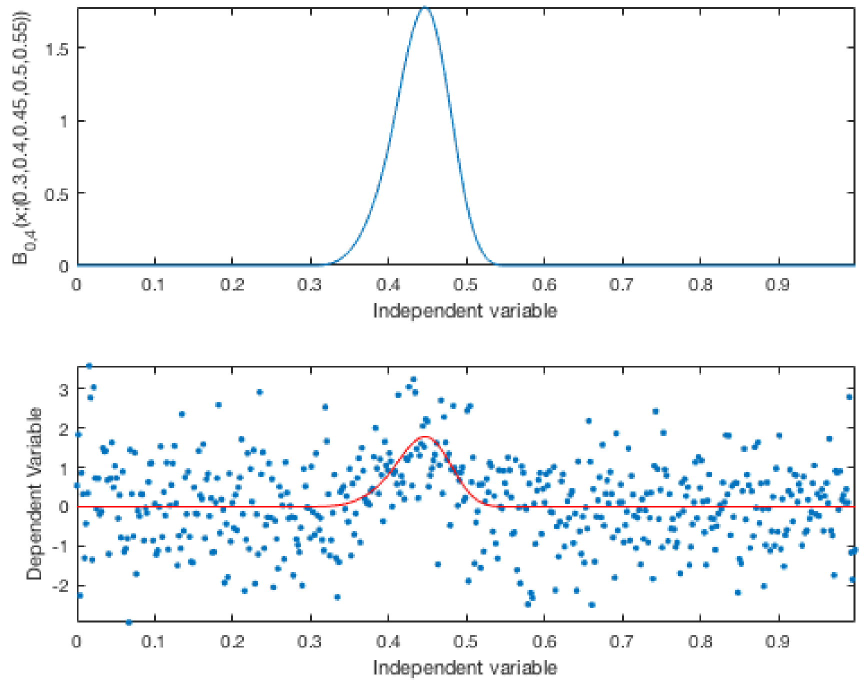
- True signal:

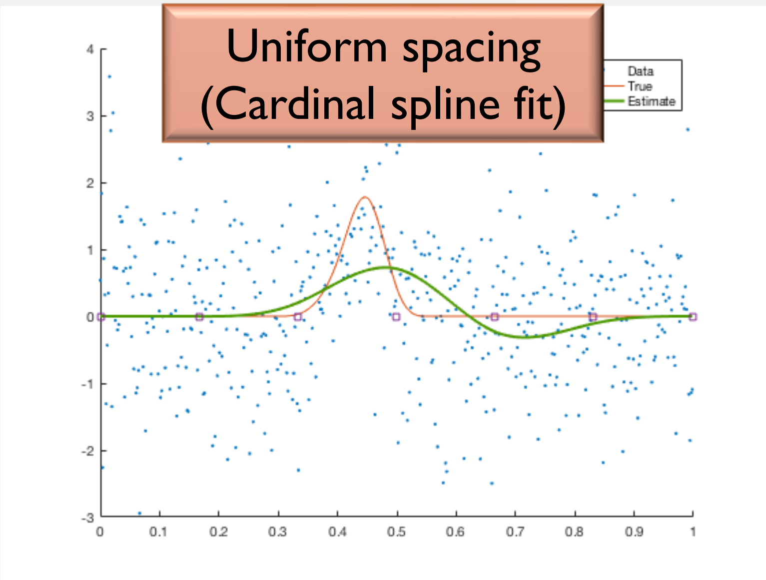
$$f(x;\bar{\theta}) = 10 \times B_{0,4}(x;\bar{c});$$
$$\bar{c} : \text{Breakpoint locations}$$
$$\bar{c} = (0.3, 0.4, 0.45, 0.5, 0.55)$$

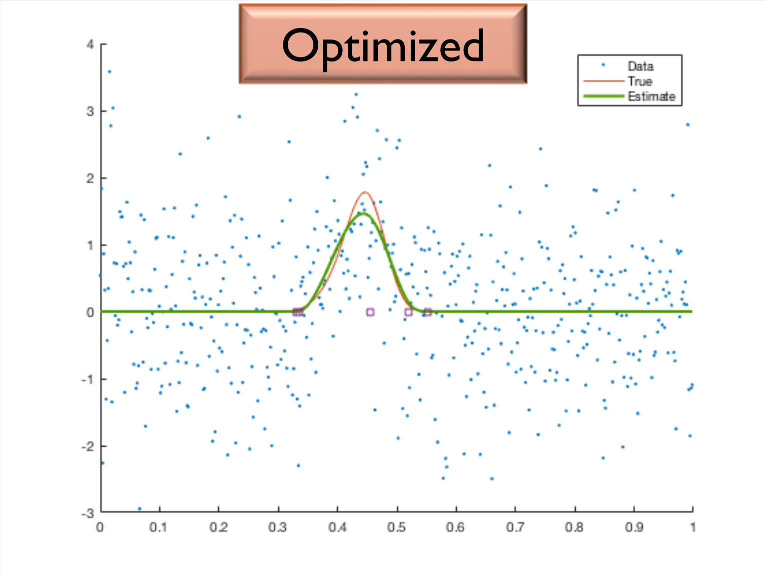
- White Gaussian Noise (WGN): iid Normal with mean $= 0$ and variance $= 1$
- 100 data realizations
- PSO Search space (after reparametrization of breakpoints):

$$\bar{b} \rightarrow \bar{\alpha}; \alpha_i \in (0,1)$$

*True breakpoints not uniformly spaced ⇒ not centered in search space







FIXED NUMBER OF BREAKPOINTS

- The true signal has 5 breakpoints
- We keep the same number of breakpoints for the spline to be fitted
- PSO tuning:
  - $N_{runs} = 4$
  - $N_{iter} = 200$
- PSO optimized breakpoints show much better performance than uniformly spaced breakpoints

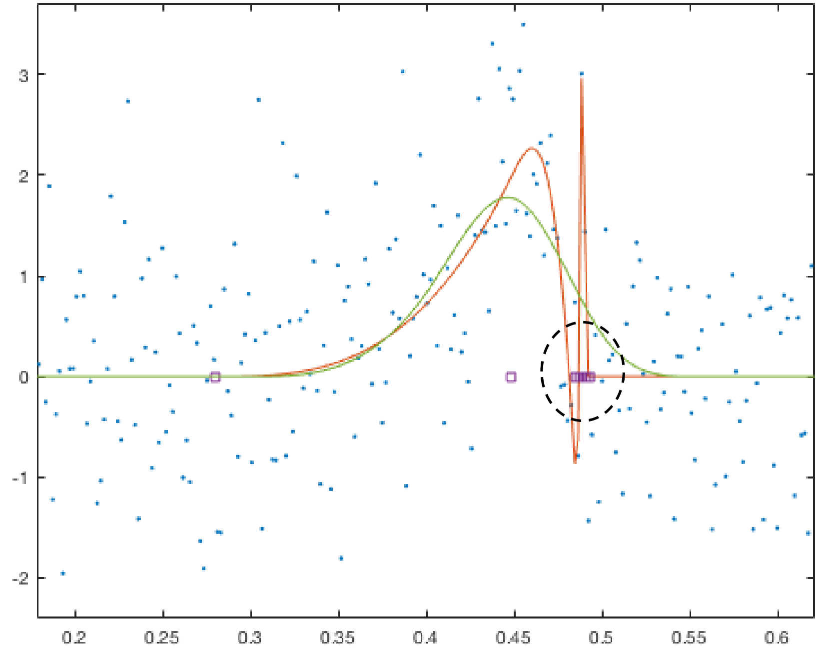
VARIABLE NUMBER OF BREAKPOINTS

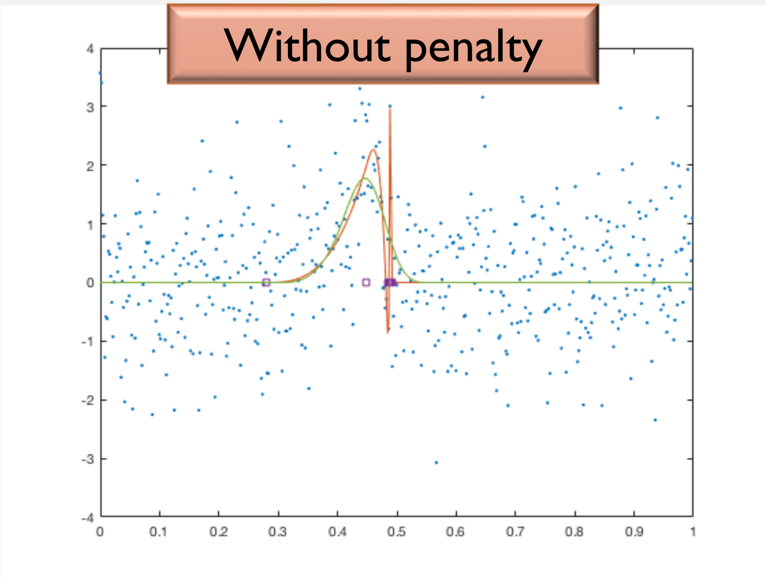
- In a realistic application, we do not have prior knowledge of the number of breakpoints to use
- $\Rightarrow$ Use **model selection**:
  - Fit the data with different breakpoint numbers ($\equiv$ different models)
  - Select the best number of breakpoints using the Akaike Information Criterion (AIC)
- *Model selection is a vast topic and AIC is not the only approach

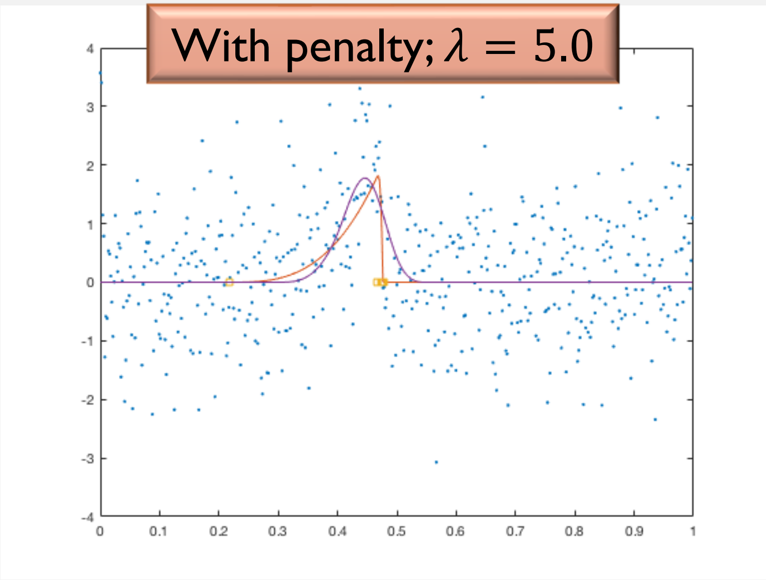
MODEL SELECTION

<u>Example</u>

- Best model found has 7 breakpoints
  - 5 breakpoints in true signal
- Variable number of breakpoints ⇒ Excessive freedom in the model
- Conspiracy: Excess knots cluster and coefficients $\bar{\alpha}$ increase to fit outliers
- Smoothness constraint on solution is violated







PENALIZED SPLINE

- Penalized spline fit: Add a penalty term (regulator) to the cost function

$$\min_{\bar{b}}\left(\min_{\bar{\alpha}}\left(\sum_{i=0}^{N-1}\left(y_i-\sum_{j=0}^{M-1}\alpha_j B_{j,4}(x_i;\bar{b})\right)^2+\lambda\sum_{j=0}^{M-1}\alpha_j^2\right)\right)$$

- λ: Regulator gain
- Penalize solutions that have large values of α_i

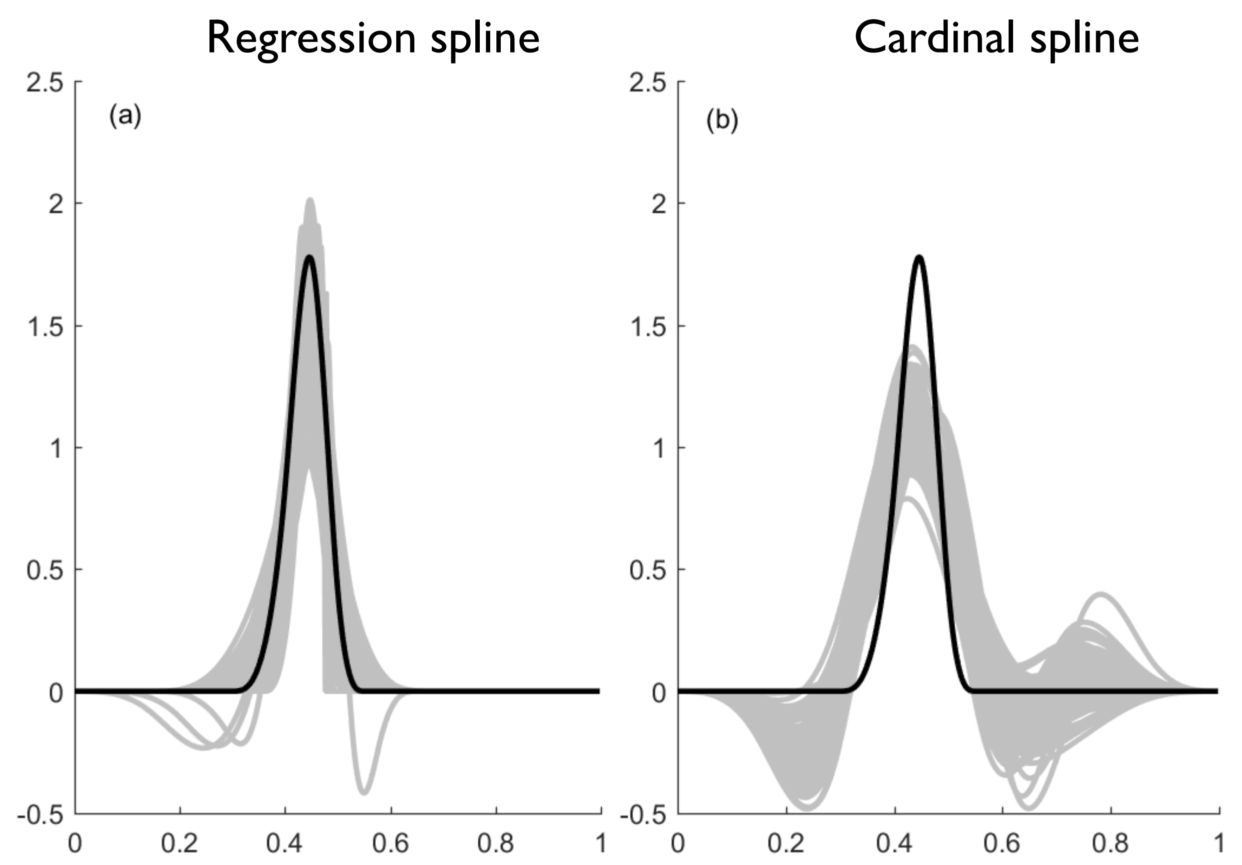
REGULARIZATION AND MODEL SELECTION

Cardinal spline fit

- Breakpoints spaced uniformly
- Model selection used

Regression spline fit

- Breakpoints optimized by PSO
- Model selection used
- Penalized spline used

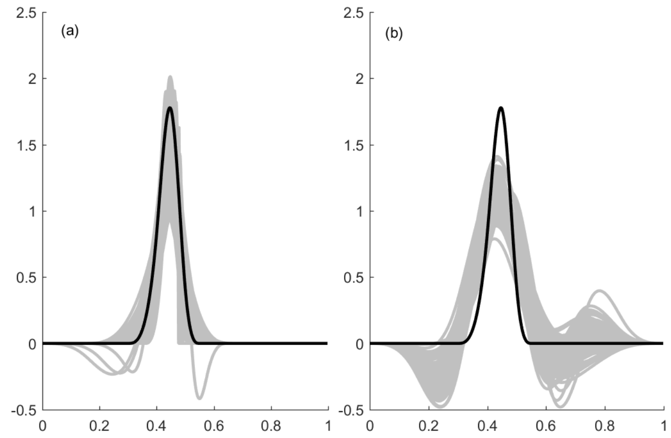


- 100 data realizations
- Breakpoint numbers: $\{5,6,7,8,9\}$

SUMMARY

BREAKING THE OPTIMIZATION BARRIER

- Non-parametric regression
  - Linear model
    - Optimization: simple
  - Non-linear model
    - Optimization: Challenging
      - Swarm intelligence

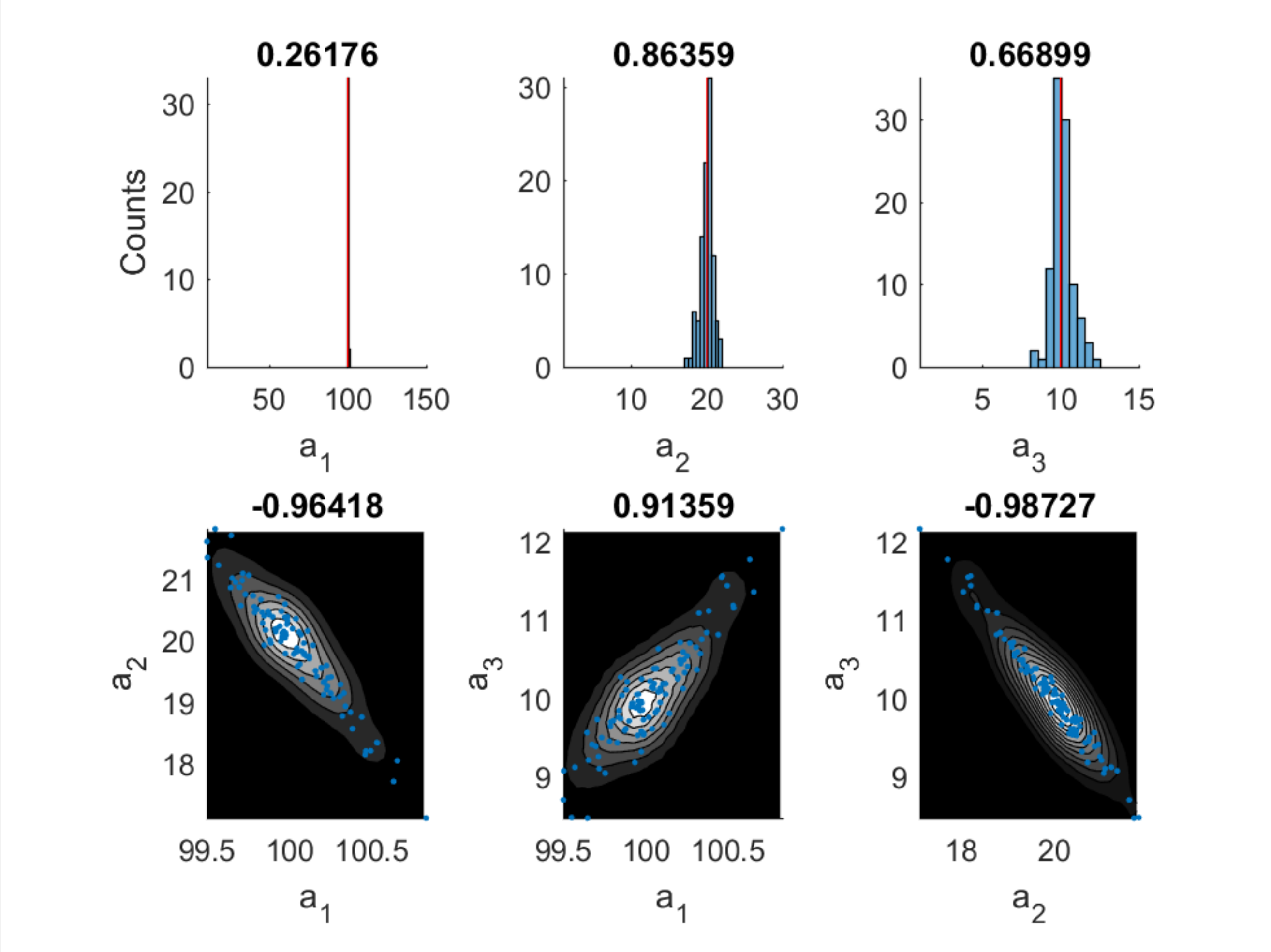


Non-linear model fit by PSO

linear model fit

BREAKING THE OPTIMIZATION BARRIER

- Parametric regression
- Non-linear model
- Optimization: Challenging
- Swarm intelligence



SWARM INTELLIGENCE AND BIG DATA

Big data era: datasets and inference problems have become more complex

Flexible modeling ⇒ Non-parametric regression methods ⇒ large number of parameters ⇒ Optimization bottleck

- Forced to use linear models but non-linear models may be better

Swarm intelligence methods like PSO should be in the toolbox of every big data analyst

- Not a magic pill! Try SI methods on simpler problems first