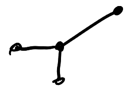
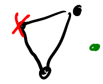
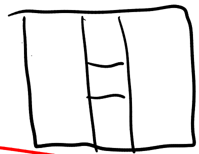
Last Time: Direct / Pattern Search

- Coordinate Search
- Generalized Patern Search
- Nelder-Mead Simplex Method
- DIRECT (Lipschitz-Inspired)







Today: Stochastic + Population Methods

- Simulated Annealing
- Cross-Entropy

2 challenges

1. Finding local min within neighborhood ↙ so far
2. Finding neighborhood with lowest min

Simulated Annealing

temperature decay schedule

initialize x hyperparameters D, $t(k)$

loop

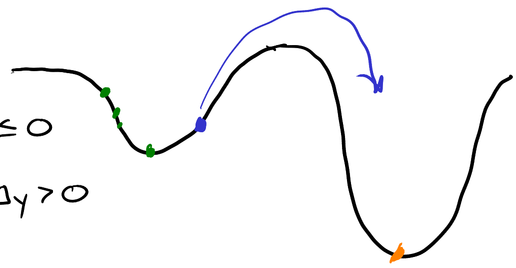
$x' \leftarrow x + \text{rand}(D)$

$\Delta y \leftarrow f(x') - f(x)$

$\text{accept} \leftarrow \text{rand}(U(0,1)) \leq \begin{cases} 1 & \text{if } \Delta y \leq 0 \\ e^{-\Delta y / t(k)} & \text{if } \Delta y > 0 \end{cases}$

if accept

$x \leftarrow x'$



Common temperature schedules

$t^k = t^1 \ln(2) / \ln(k+1)$ "logarithmic"   asymptotically optimal, very slow

$t^{k+1} = \gamma t^k$ "exponential"

$t^k = \frac{t^1}{k}$ "fast annealing"

Cross-Entropy Method

Parameterized Distribution $D(\theta)$

initialize θ ← Gaussian: μ, Σ hyperparameters: D, m, m_{elite}

loop

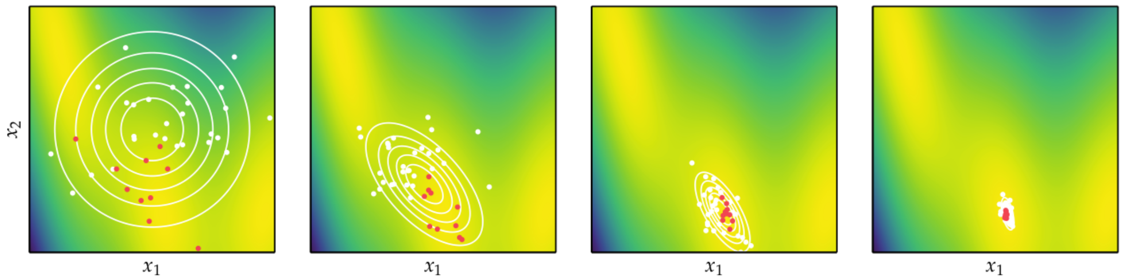
population ← sample m from $D(\theta)$

elite ← select m_{elite} best samples

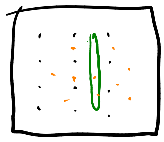
θ ← fit to elite ↖ minimize "cross entropy" fit: If D is Gaussian $\quad \theta = (\mu, \Sigma)$

$$\vec{\mu}^{k+1} \leftarrow \frac{1}{m_{elite}} \sum_{i=1}^{m_{elite}} \vec{x}^i$$

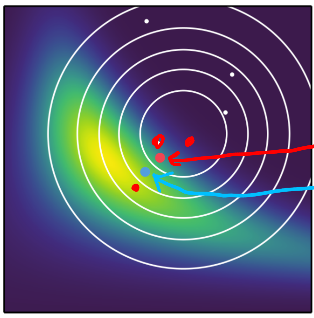
$$\Sigma^{k+1} \leftarrow \frac{1}{m_{elite}} \sum_{i=1}^{m_{elite}} (\vec{x}^i - \vec{\mu}^{k+1})(\vec{x}^i - \vec{\mu}^{k+1})^T$$



Other Stochastic Algorithms



- Mesh-adaptive direct search ← stochastic pattern search
- Natural Evolution Strategies
  estimate $\nabla_\theta E[f(\vec{x})]$ rather than fitting θ to elite
- Covariance Matrix Adaptation
  - similar to cross-entropy, but weight elite samples



Cross-entropy

cov. matrix adaptation

Population Methods

Genetic Algorithms

Chromosome: vector of bits / real numbers

[0, 1, 1, 0 . . .]

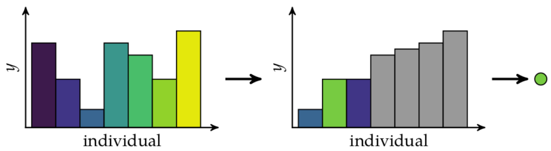
[5.2, 6.3 . . .] $\vec{x}$

← in practice

Loop
- Selection
- Crossover
- Mutation

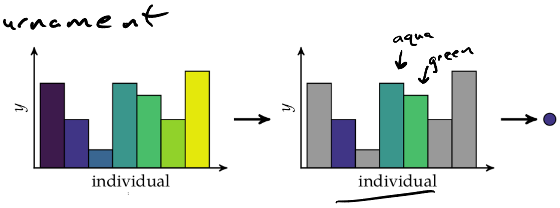
Selection

Truncation



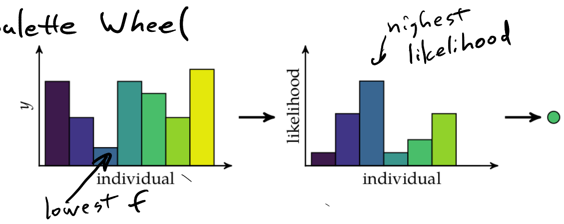
select best k individuals
randomly select from best k

Tournament



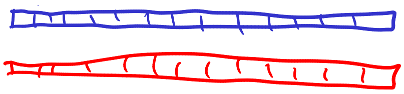
randomly select k
take best

Roulette Wheel

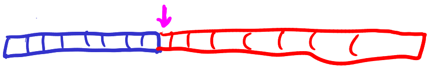


Crossover

Parent 1
Parent 2



Single Point
Child



Two Point

Uniform

Mutation

Bitwise: each bit has an independent probability of flipping
Gaussian: each element has Gaussian noise added to it

Particle Swarm

initialize population $\{{}^i\vec{x}\}_{i=1}^{m}$ hyperparams: w, c_1, c_2

loop

for $i \in 1..m$

${}^i\vec{x} \leftarrow {}^i\vec{x} + {}^i\vec{v}$

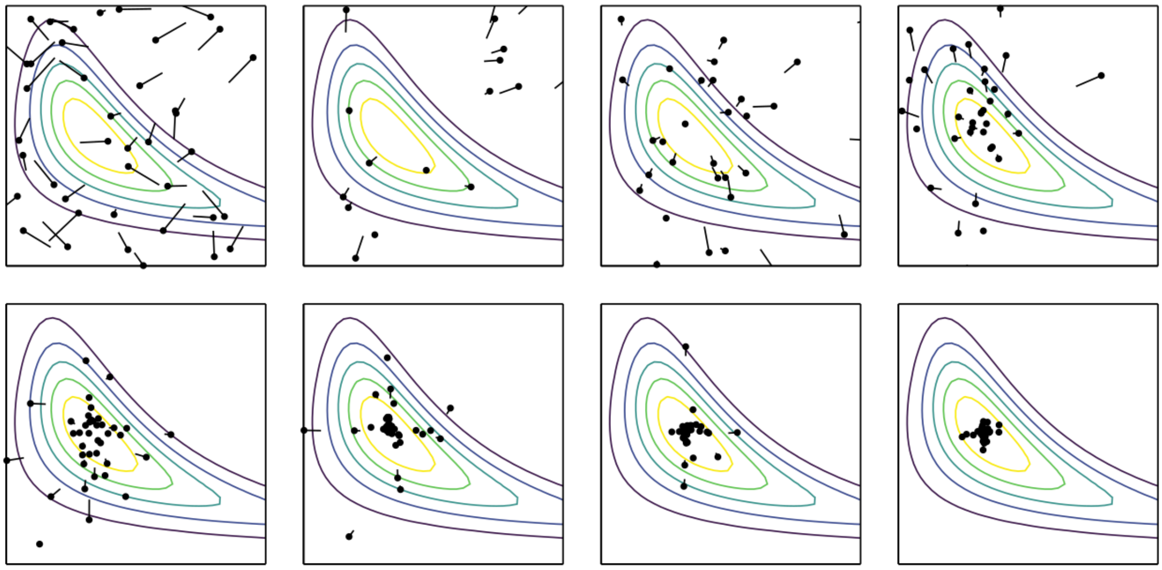
${}^i v \leftarrow w\,{}^i v + c_1 r_1 ({}^i\vec{x}_{best} - {}^i\vec{x}) + c_2 r_2 (\vec{x}_{best} - {}^i\vec{x})$

if $f({}^i\vec{x}) < f({}^i\vec{x}_{best})$

${}^i\vec{x}_{best} \leftarrow {}^i\vec{x}$

rand$(U(1,0))$

update $\vec{x}_{best}$



Firefly Algorithm

$\vec{a}$ moves toward $\vec{b}$

$$\vec{a} \longleftarrow \vec{a} + \beta I\left(\underline{\|\vec{b}-\vec{a}\|}\right)\left(\vec{b}-\vec{a}\right) + \alpha\,\varepsilon$$

intensity

Gaussian

hyperparameters

$$I(r) = \frac{1}{r^2}$$

$$I(r) = e^{-\gamma r}$$

$$I(r) = e^{-\gamma r^2}$$