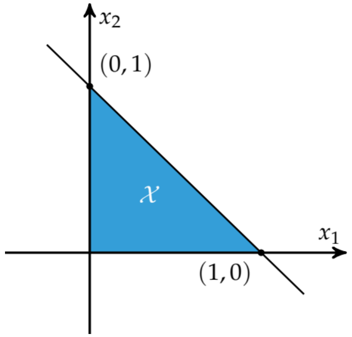
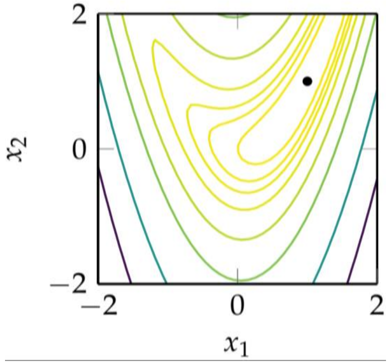
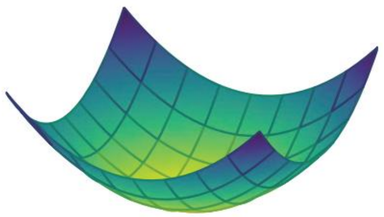
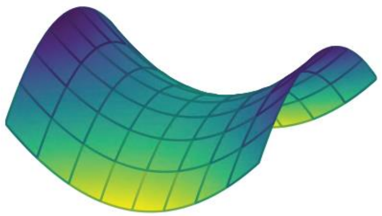
Tecnológico Nacional de México
Instituto Tecnológico de Orizaba/Celaya

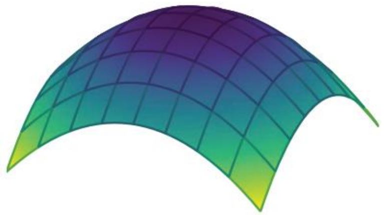
Modern Computing Algorithms for Process Optimization with Julia Programming











"8 – Stochastic Methods"

By:

Dr. Kelvyn Baruc Sánchez Sánchez
Postdoctoral Researcher/I.T. Celaya

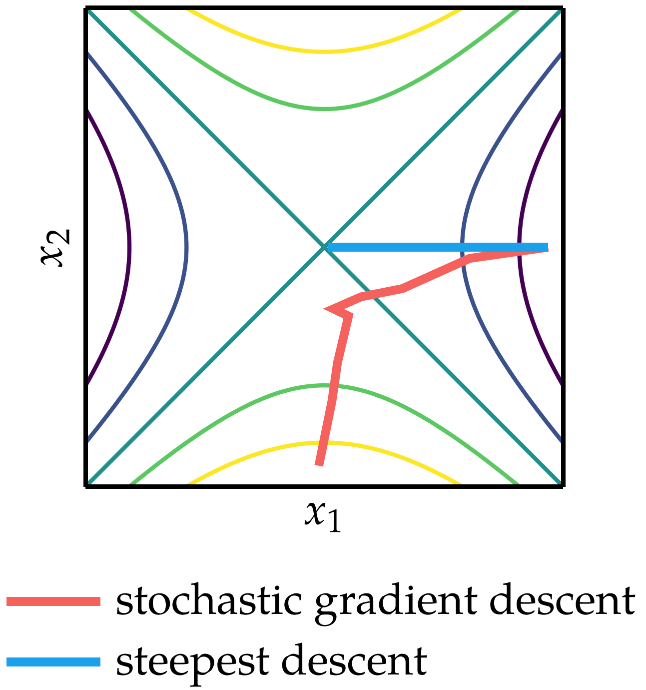
Stochastic Methods

- Employ randomness strategically to help explore design space
- Randomness can help escape local minima
- Increases chance of searching near the global minimum
- Typically rely on pseudo-random number generators to ensure repeatability

Noisy Descent

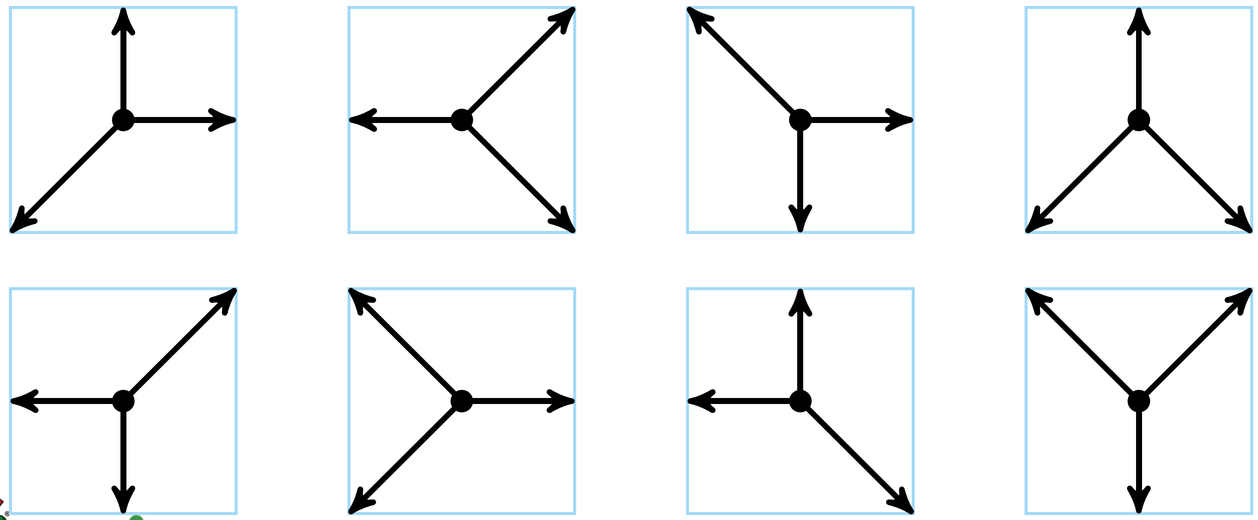
- Gradient descent with additional random noise term
- Step size must meet certain conditions to guarantee convergence

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \alpha \mathbf{g}^{(k)} + \boldsymbol{\epsilon}^{(k)}$$

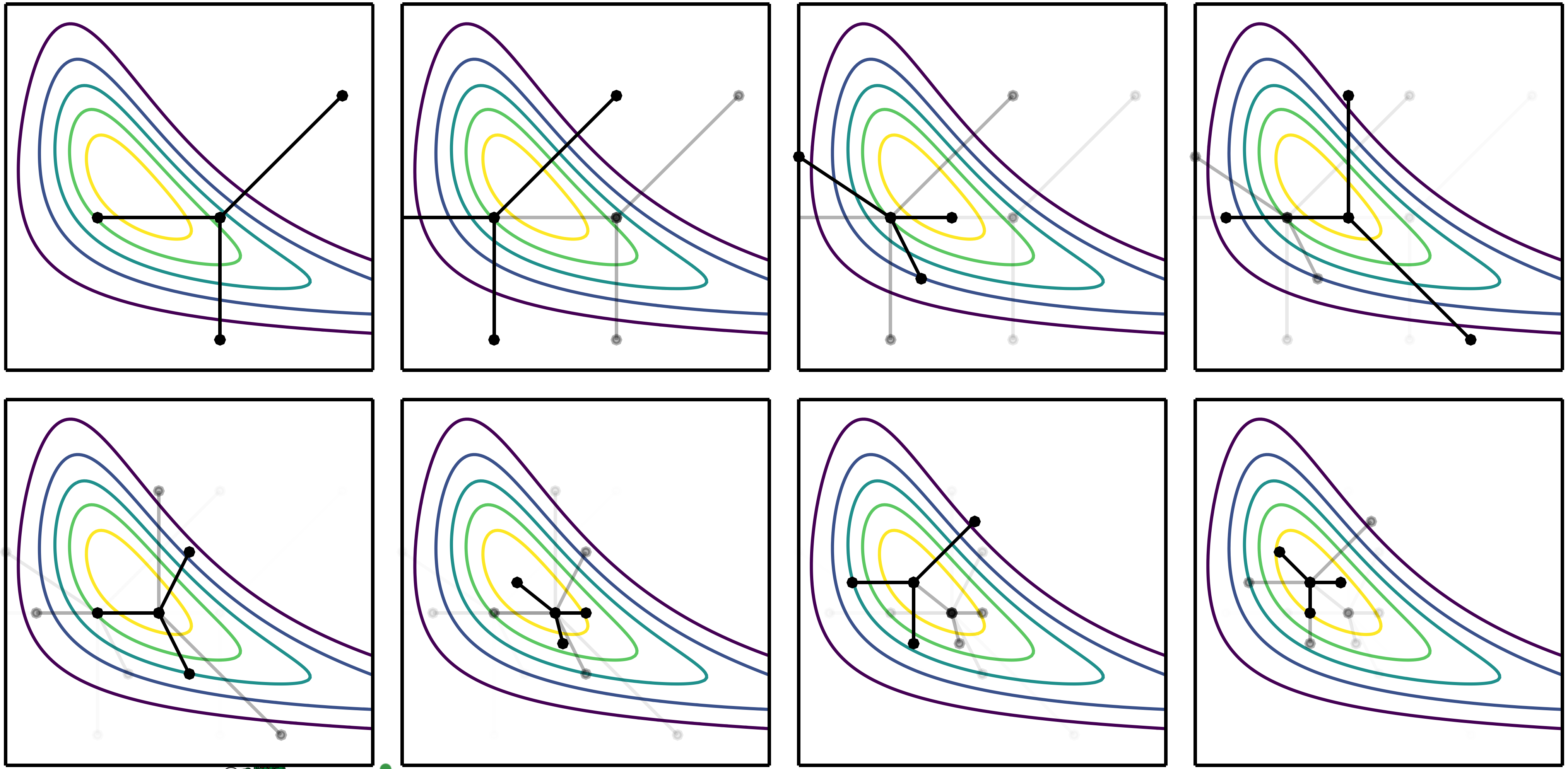


Mesh Adaptive Direct Search

- Similar to generalized pattern search method, but searches in a random set of directions that span the search space
- Example: set of positive spanning sets constructed from nonzero directions $d_1, d_2 \in \{-1,0,1\}$



Mesh Adaptive Direct Search



Simulated Annealing

- Intelligently controls the degree of randomness added to stochastic search methods
- Randomness is analogous to temperature in metallurgy
- Initially, the randomness added to function evaluations is large
- The “temperature” is then slowly lowered according to a predetermined “annealing schedule”

Simulated Annealing

- Each step must then be “accepted” as the new design point with probability defined by the Metropolis criterion, helping the algorithm escape local minima
- Metropolis criterion probability of acceptance

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

Simulated Annealing

Annealing schedules

- Logarithmic annealing schedule

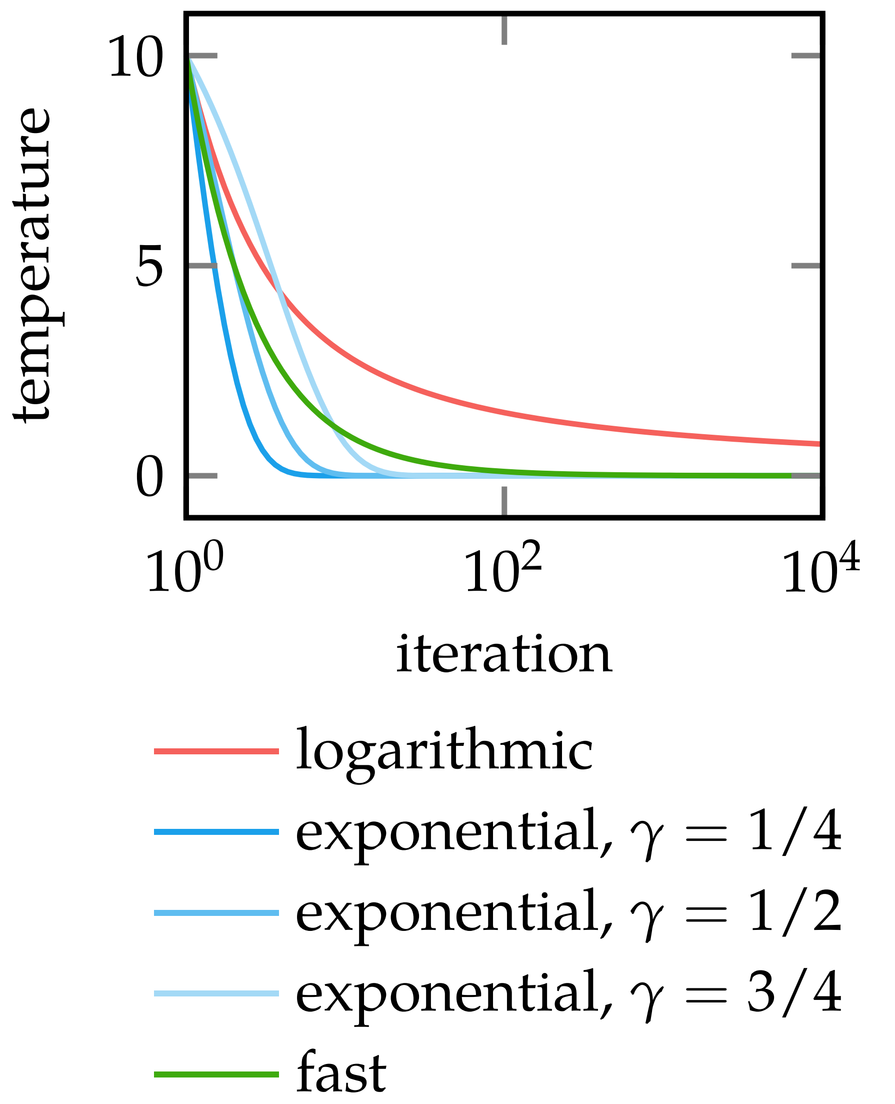
$$t^{(k)} = t^{(1)} \frac{\ln(2)}{\ln(k+1)}$$

- Exponential annealing schedule

$$t^{(k+1)} = \gamma t^{(k)}$$

- Fast annealing

$$t^{(k)} = \frac{t^{(1)}}{k}$$



Dr. Kelvyn B. Sánchez Sánchez
Instituto Tecnológico de Celaya

Simulated Annealing

- Corana et al. 1987 introduced variable step-size
- First, a cycle of random moves is performed in each direction

$$\mathbf{x}' = \mathbf{x} + r v_i \mathbf{e}_i$$

where r is randomly sampled from [-1,1]

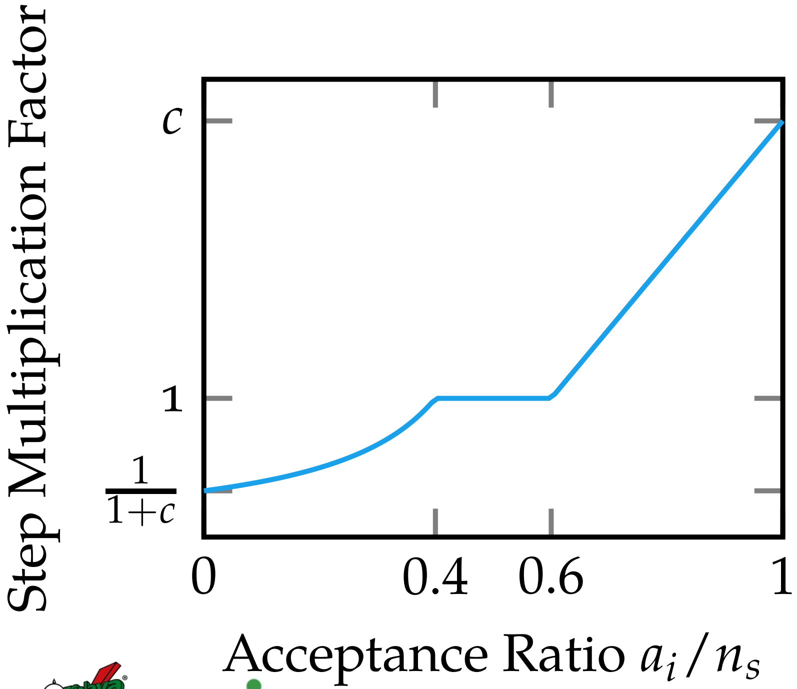
- After n_s cycles, step size is adjusted according to

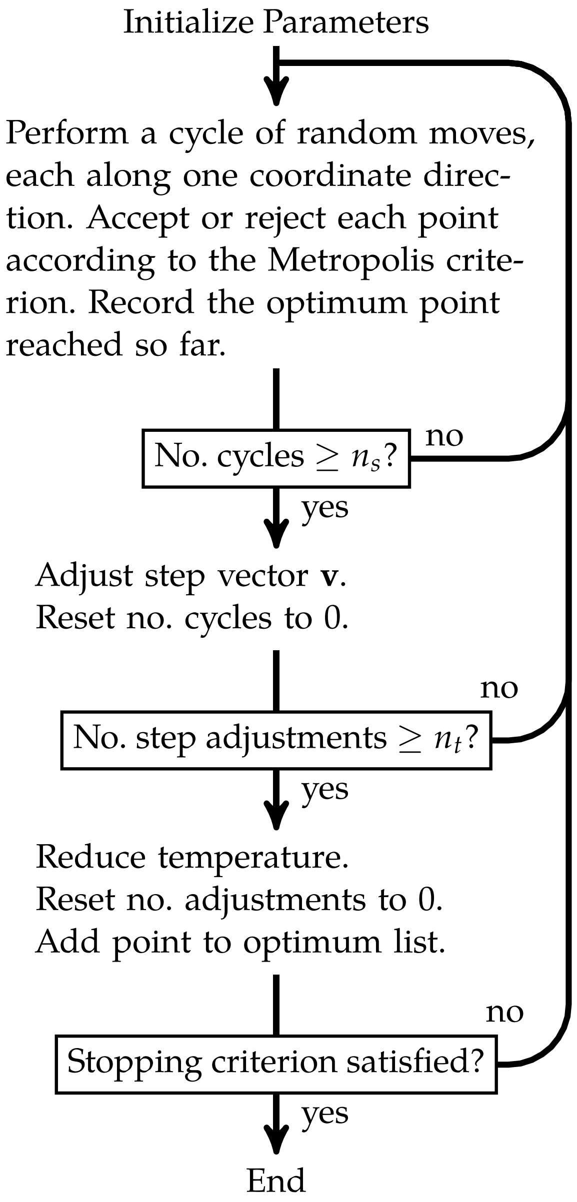
$$v_i = \begin{cases} v_i\left(1 + c_i \frac{a_i/n_s - 0.6}{0.4}\right) & \text{if } a_i > 0.6 n_s \\ v_i\left(1 + c_i \frac{0.4 - a_i/n_s}{0.4}\right)^{-1} & \text{if } a_i < 0.4 n_s \\ v_i & \text{otherwise} \end{cases}$$

where c is typically set to 2

Simulated Annealing

- The variable step size introduced by Corana et al. 1987 regulates the ratio of accepted-to-rejected points to about 50%

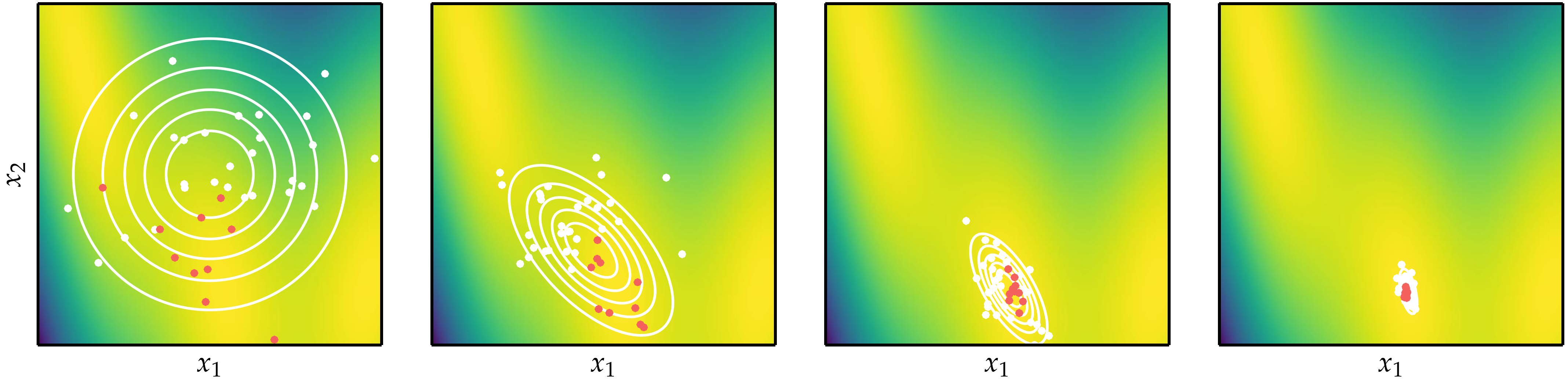




Cross-Entropy Method

- Maintains explicit probability distribution over design space often called a proposal distribution
- Requires choosing a family of parameterized distributions
- At each iteration, a set of design points are sampled from the proposal distribution; these are evaluated and ranked
- The best-performing subset of samples, called elite samples, are retained
- The proposal distribution parameters are then updated based on the elite samples, and the next iteration begins

Cross-Entropy Method




```julia
using Distributions
function cross_entropy_method(f, P, k_max, m=100, m_elite=10)
    for k in 1 : k_max
        samples = rand(P, m)
        order = sortperm([f(samples[:,i]) for i in 1:m])
        P = fit(typeof(P), samples[:,order[1:m_elite]])
    end
    return P
end
```

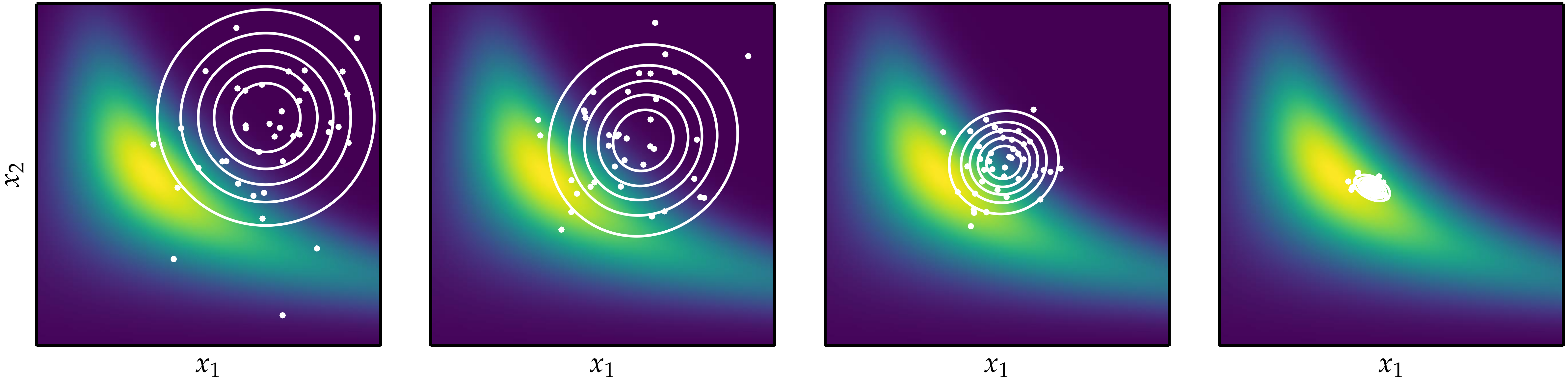
Dr. Kelvyn B. Sánchez Sánchez
Instituto Tecnológico de Celaya

Natural Evolution Strategies

- Similar to cross-entropy method, except instead of parameterizing distribution based on elite samples, it is optimized using gradient descent
- The distribution parameter gradient is estimated from the set of function evaluations

```
using Distributions
function natural_evolution_strategies(f, θ, k_max; m=100, α=0.01)
    for k in 1 : k_max
        samples = [rand(θ) for i in 1 : m]
        θ -= α*sum(f(x)*∇logp(x, θ) for x in samples)/m
    end
    return θ
end
```

Natural Evolution Strategies



Covariance Matrix Adaptation

- Same approach as natural evolution strategy and cross entropy method, but the proposal distribution is a multivariate Gaussian parameterized by a covariance matrix
- Sample efficient

Summary

- Stochastic methods employ random numbers during the optimization process
- Simulated annealing uses a temperature that controls random exploration and which is reduced over time to converge on a local minimum
- The cross-entropy method and evolution strategies maintain proposal distributions from which they sample in order to inform updates

Summary

- Natural evolution strategies uses gradient descent with respect to the log likelihood to update its proposal distribution
- Covariance matrix adaptation is a robust and sample-efficient optimizer that maintains a multivariate Gaussian proposal distribution with a full covariance matrix