Stochastic Optimization

Covariance Matrix Adaptation (CMA-ES)

Outline

1. **Optimization problem**
2. **Why the search may be difficult?**
   - The pros and cons of stochastic search
3. **The stochastic search**
   - The most general template
   - The most simple Gaussian search
   - Elaborate Gaussian search
4. **Some general concepts to understand**
   - The multivariate distribution
   - Selection and recombination *(*Mean ***m*** update*)*
   - Step size control (Step-size **σ** update)
   - The rank-1 update (Covariance matrix **C** update)
   - The rank-μ update (Covariance matrix **C** update)
5. **CMA-ES : The Algorithm**
6. **Synthetic Groundwater flow model**
   - Precision of the estimated covariance matrix
   - Covariance matrix Monte Carlo simulation
   - Parallel performance of CMA-ES
7. **Baton Rouge groundwater flow models**

1. The optimization problem

Types of optimization problems

Problem

Let $x \in ¡^n, f(x) : ¡^n \to ¡$,

$$\text{find } \min_x f(x),$$

where you evaluate $f(x)$ for any $x \in ¡^n$, and x is a set of decision variables with size n

2nd order : $\nabla^2 f(x)$ can be evaluated

Gradient : $\nabla f(x)$ can be evaluated

Blackbox : only $f(x)$ can be evaluated

Blackbox: generally stochastic search where one needs to learn from the history in some way or another to have an idea where continue to search

A classification of optimization algorithms

Class	Example
Local derivative	Levenberg-Marquardt
Global heuristic	Simulated Annealing / Genetic algorithm
Global heuristic-local derivative	Genetic algorithm and Levenberg-Marquardt
Local-global heuristic	CMA-ES

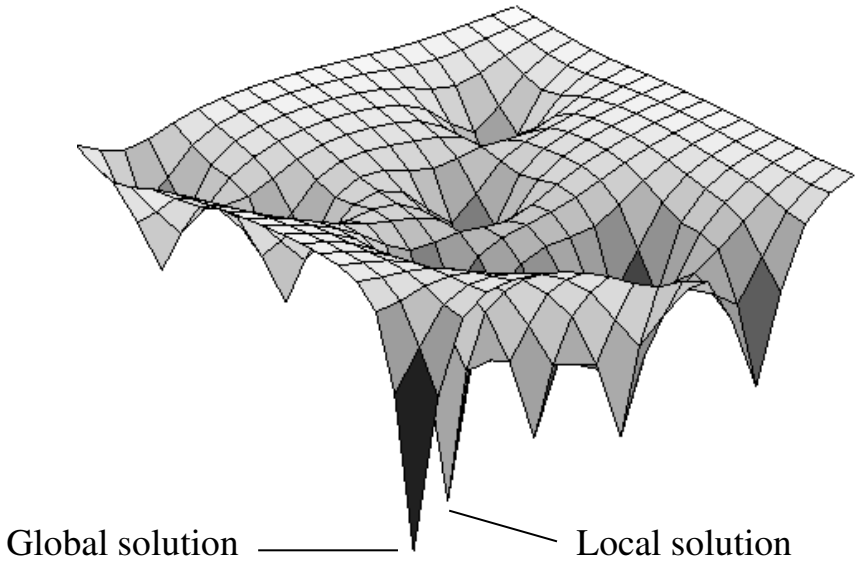
Further classifications for heuristic search	Example
Singe solution vs. Population of solutions	Simulated Annealing vs. Differential Evolution
Distribution estimation vs. Constructive	CMA-ES vs. Differential Evolution

When do we need heuristic search?

Non-derivative algorithms

- Only $f(x)$ can be evaluated
- $\nabla f(x)$ or $\nabla^2 f(x)$ are not directly accessible
- Or when derivative search not successful

$$f(x_1,...,x_n) = \sum_{i=1}^{m} \frac{1}{a_i + \sum_{j=1}^{n} (x_j - b_{ij})^2}$$



2. Why the search may be difficult ?

Or why do we need stochastic search

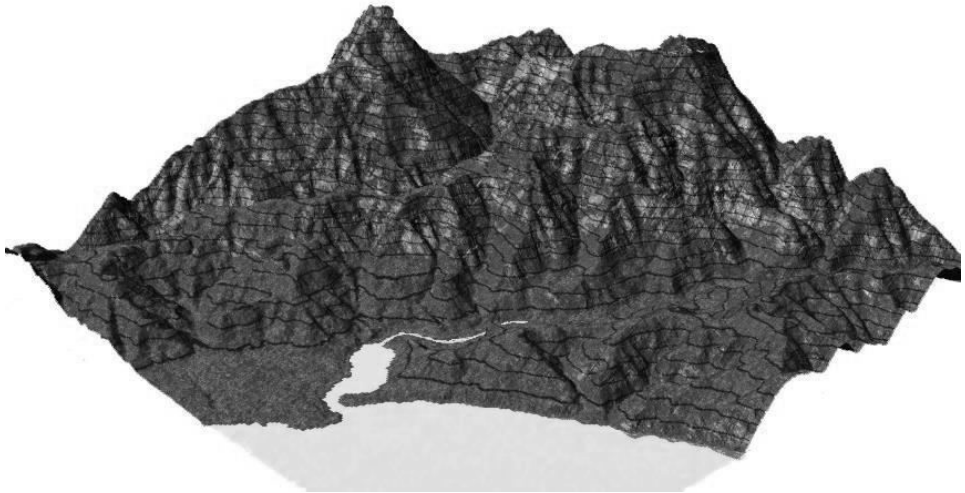
Or the pros and cons of stochastic search

1. Nonlinearity

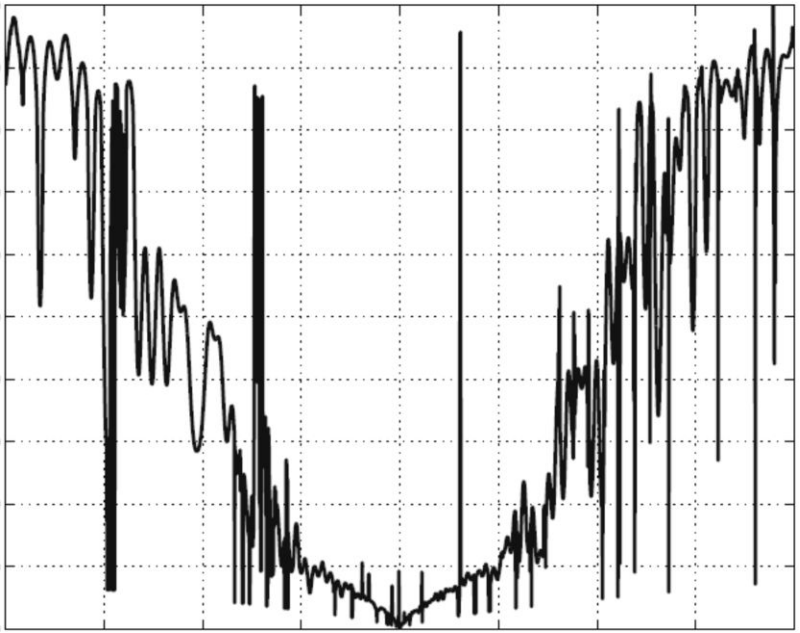
Sources of nonlinearity

- PDE describing unconfined flow is non-linear $\partial h/\partial t \alpha h \nabla h$
- Nonlinearity due to high recharge that causes step gradients
- Nonlinearity due heterogeneous K field
- Nonlinearity due to nonlinear interphase mass transfer and (bio)geochemical transformations

2. Ruggedness



http://www.princeton.edu/~oa/manual/mapcompass.shtml



Auger and Hansen (2013)

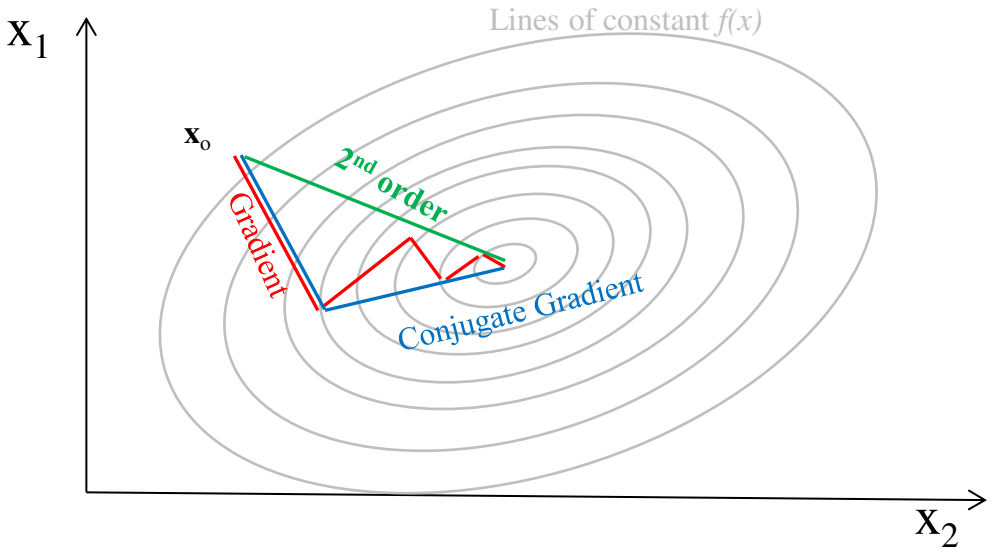
Cut from a 5-D example, (easily) solvable with non-derivative algorithms

Non-smooth, discontinuous and multimodal

3. Ill-conditioning

Ill-conditioning means squeezed level sets (high curvature):

For a curvature matrix (Hessian matrix) $\mathbf{H}_{ij} = \partial^2 f(x) / \partial x_i \partial x_j$, if the eigenvalues λ_n span a wide range of values such that the largest and the smallest eigenvalues are $\lambda_{\max} / \lambda_{\min}$ is large then the surfaces of constant $f(x)$ are very elongated

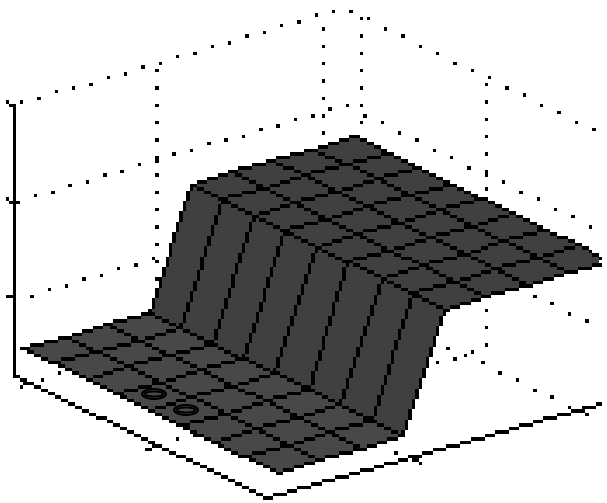


If $\mathbf{H} \approx \mathbf{I}$ (small condition number of $\mathbf{H}$) first order information $\nabla f(x)$ is sufficient. Otherwise, second order information $\nabla^2 f(x)$ is necessary

4. Noise

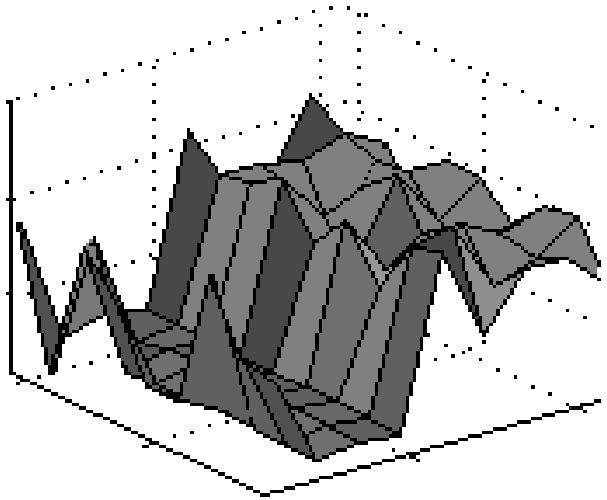
Unimportant information to the objective function conceals the useful information needed for the optimization process

True K field [m/s]



Injection wells

PSO K field [m/s]



Fooled by information from the injection wells

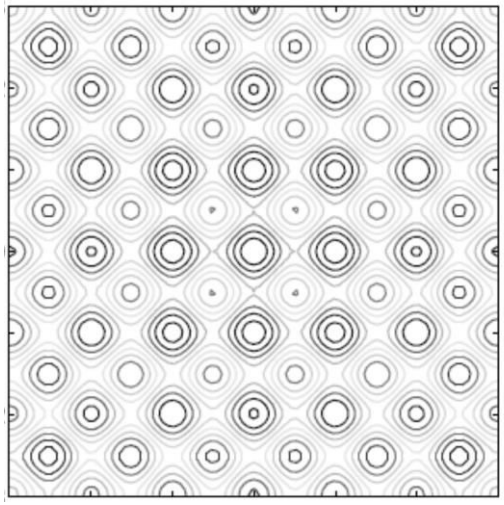
5. Non-separability

Additively decomposable Rastrigin function

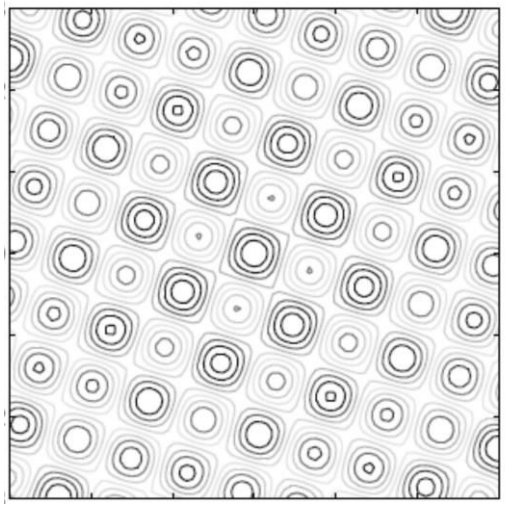
$$f(x_1,...,x_n) = \sum_{i=1}^{n} f_i(x_i)$$

f can be optimized in a sequence of n independent 1-D optimization processes

Building a non-separable problem from a separable one



$\mathbf{R}$ rotation matrix



Auger and Hansen (2013)

$f : x$ a $f(x)$ separable

$f : x$ a $f(\mathbf{R}x)$ non-separable

6. Curse of dimensionality

Rapid increase in volume associated with adding extra dimensions to a (mathematical) space

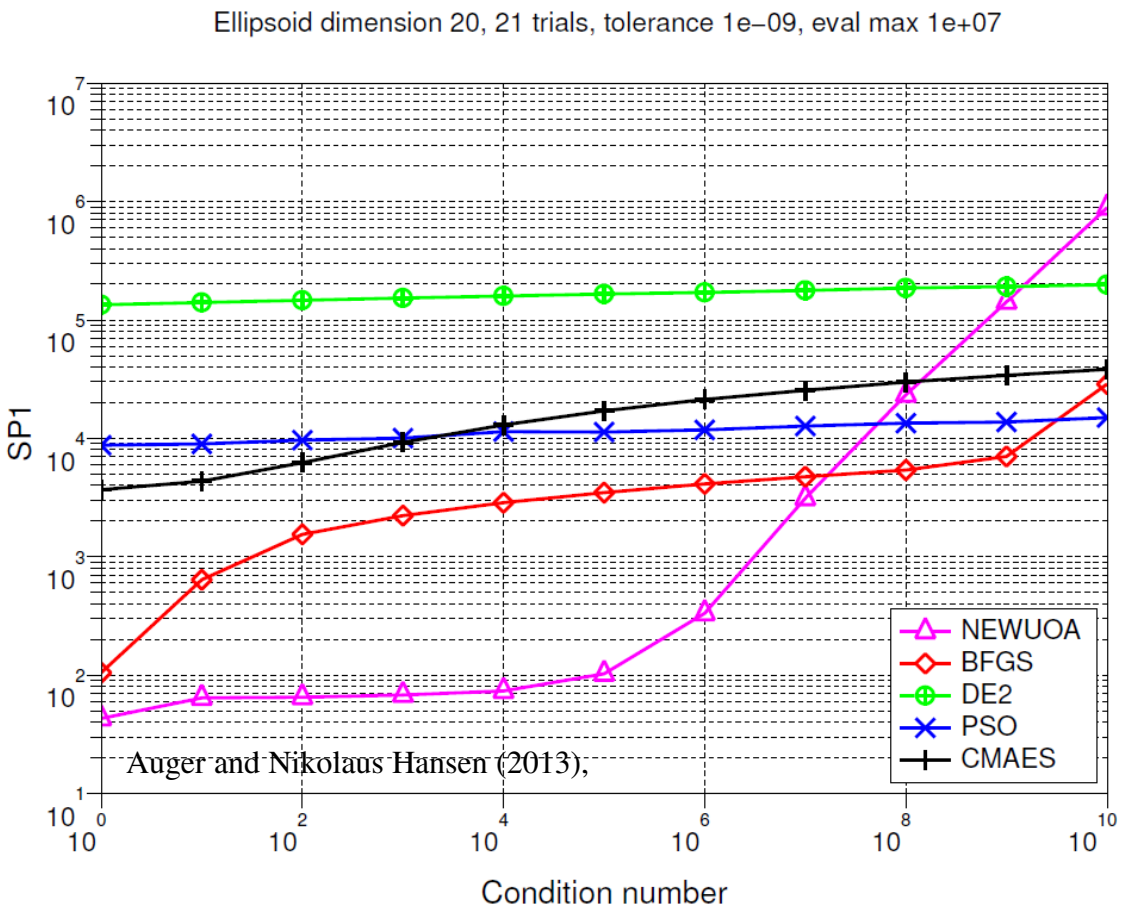
$$V = x^n$$

Example: Consider placing 100 points onto a real interval, say [0; 1]. Now consider the 10-dimensional space $[0; 1]^{10}$. To get **similar coverage** in terms of distance between adjacent points would require $100^{10} = 10^{20}$ points. A 100 points appear now as isolated points in a vast empty space.

Number of iterations $O(n^n)$
Number of iterations $O(n^2)$
Number of iterations $O(n)$

Illustration

f convex quadratic, separable with varying condition number



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with

$\boldsymbol{H}$ diagonal

g identity (for BFGS and NEWUOA)

g any order-preserving = strictly increasing function (for all other)

SP: average number of objective function evaluations to reach the target function value

BFGS: derivative (quasi-Newton)

NEWUAO: derivative free (non-stochastic)

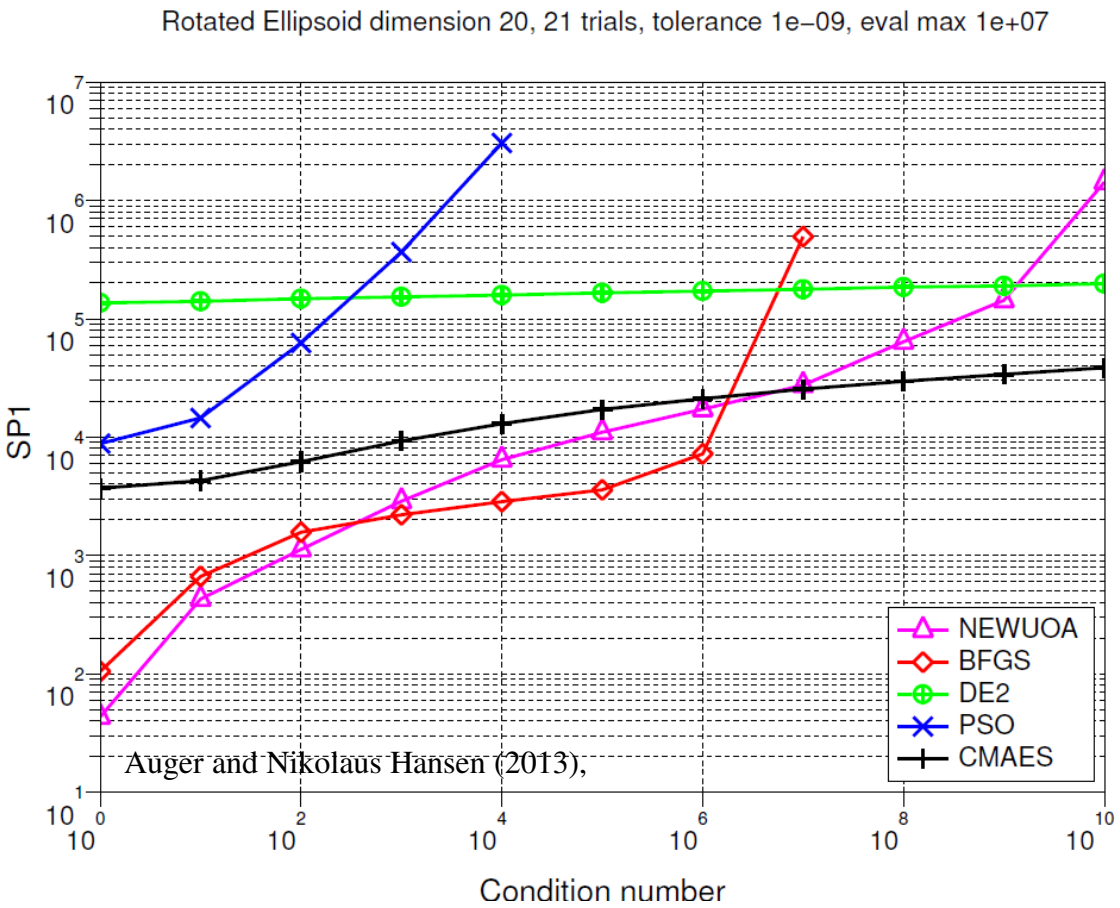
DE: stochastic (Constructive)

PSO: stochastic (Constructive)

CMA-ES (Distribution estimation)

Illustration

f convex quadratic, non-separable (rotated) with varying condition number



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with

$\boldsymbol{H}$ full

g identity (for BFGS and NEWUOA)

g any order-preserving = strictly increasing function (for all other)

SP: average number of objective function evaluations to reach the target function value

BFGS: derivative (quasi-Newton)

NEWUAO: derivative free (non-stochastic)

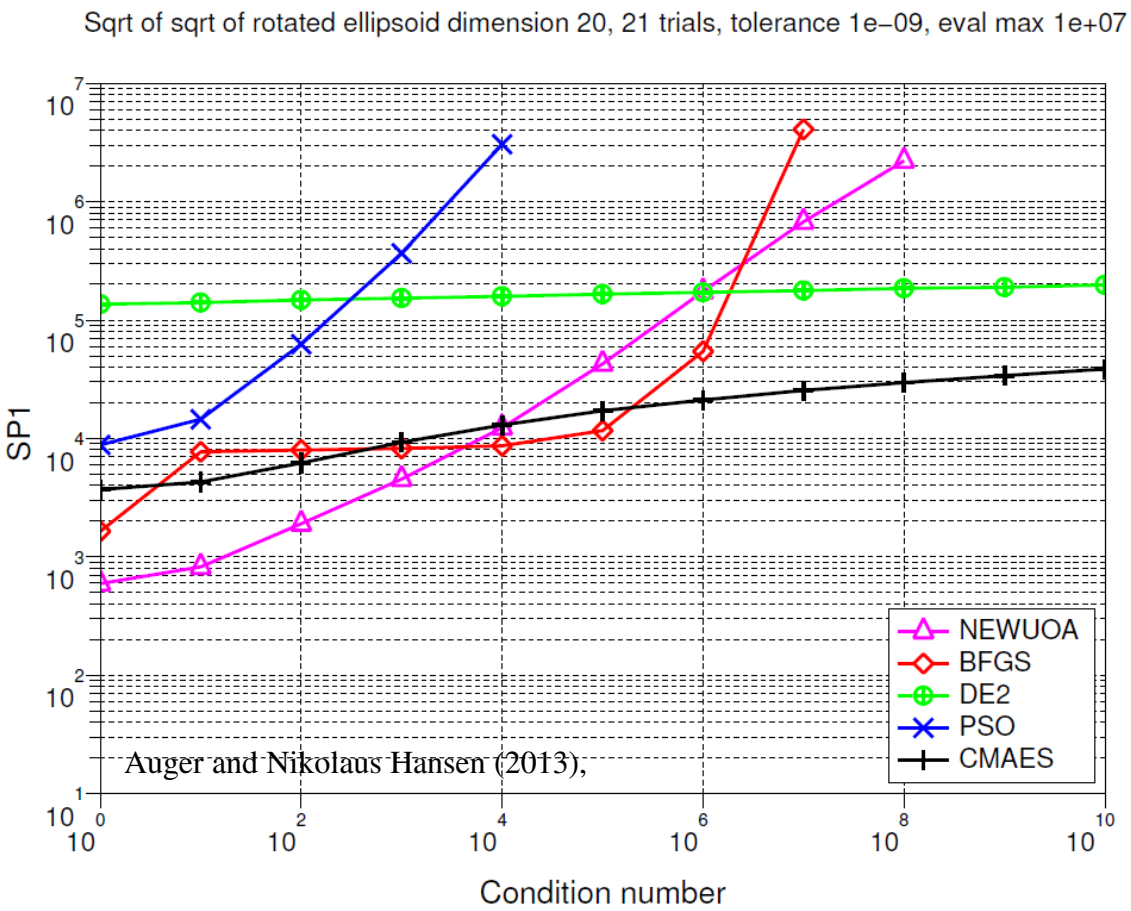
DE: stochastic (Constructive)

PSO: stochastic (Constructive)

CMA-ES (Distribution estimation)

Illustration

f non-convex, non-separable (rotated) with varying condition number



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with

$\boldsymbol{H}$ full

$g : x \mapsto x^{1/4}$ (for BFGS and NEWUOA)

g any order-preserving = strictly increasing function (for all other)

SP: average number of objective function evaluations to reach the target function value

BFGS: derivative (quasi-Newton)

NEWUAO: derivative free (non-stochastic)

DE: stochastic (Constructive)

PSO: stochastic (Constructive)

CMA-ES (Distribution estimation)

How to deal with these search difficulties ?

Problem	Possible approach
Dimensionality	Exploiting the problem structure separability, locality/neighborhood, encoding
Ill-conditioning	Second order approach
Ruggedness	Non-local policy, large sampling width (step-size) as large as possible while preserving a reasonable convergence speed, population-based method, stochastic search, …

3. The stochastic search

- The most general template
- The most simple Gaussian search
- Elaborate Gaussian search

Stochastic search

The general recipe

The algorithm maintains a distribution model $p_\theta(\mathbf{x})$

In each iteration it takes n samples $\{\mathbf{x}_i\}_{i=1}^{\lambda} : \; p_\theta(\mathbf{x}), \; \mathbf{x}_i \in$ ¡ ^{n}

Each x_i is evaluated $\rightarrow$ data $\left\{\left(\mathbf{x}_i, f(\mathbf{x}_i)\right)\right\}_{i=1}^{\lambda}$

That data is used to update θ

Stochastic search

Input : initial parameter θ, function $f(\mathbf{x})$, distribution model $p_\theta(\mathbf{x})$, update heuristic $h(\theta, D)$

Output : final θ and best solution $\mathbf{x}$

1 : **repeat**

2 : Sample $\{\mathbf{x}_i\}_{i=1}^{\lambda} \sim p_\theta(\mathbf{x})$

3 : Evaluate samples, $D = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{\lambda}$

4 : Update $\theta \leftarrow h(\theta, D)$

5 : **Until** θ converges

Toussaint (2013)

The crucial difference between algorithms is their choice of $p_\theta(\mathbf{x})$

Stochastic search

The parameter θ is the only "knowledge/information" that is being propagated between iterations

- θ encodes what has been learned from the history
- θ defines where to search in the future

Knowledge/information θ

- Simulated Annealing: θ is the "current point" and a temperature
- Differential Evolution and Genetic Algorithm : θ is a parent population
- Evolution Strategy: θ defines a Gaussian with mean and covariance

DE and GA: $\mathbf{x} \in \{0,1\}^n$, crossover and mutation difine $p_\theta(\mathbf{x})$

ES: $\mathbf{x} \in ¡^{\,n}$, often Gaussian $p_\theta(\mathbf{x})$

Example: Gaussian search distribution

From 1960s (Rechenberg & Schwefel): simplest type of distribution model

$$\theta = (\mathbf{m}), \qquad p(x) = N(\mathbf{x} \mid \mathbf{m}, \sigma^2)$$

a n-dimensional isotropic type of distribution model

Given $D = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{\lambda}$, D′=best μ of D

Compute the new mean **m** from D′

This algorithm is called Evolution Strategy (μ , λ)-ES

- The Gaussian is meant to represent a "species"
- λ offspring are generated
- The best μ selected

The other one is the elite selection (μ + λ)-ES

$$\theta = (\mathbf{m}, \mathrm{D}'), \qquad p(x) = N(\mathbf{x} \mid \mathbf{m}, \sigma^2), \quad \mathrm{D}' = \text{best } \mu \text{ of } \mathrm{D}' \cup \mathrm{D}$$

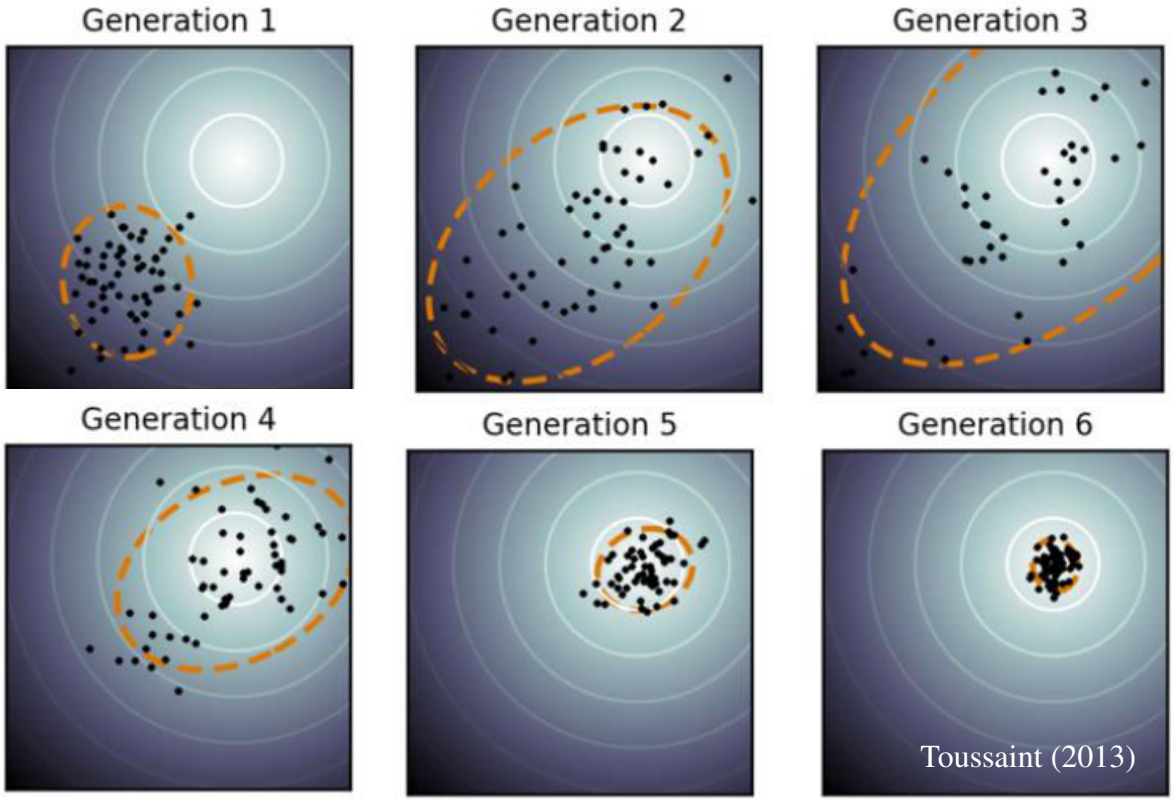
You could also have (1 + 1)-ES : Hill climber

Example: Gaussian search distribution

An obvious critique of the simple Evolution Strategies:!

– The search distribution $p(x) = N(\mathbf{x} \,|\, \mathbf{m}, \sigma^2)$ is isotropic

– The variance σ is fixed



$$\mathbf{x} \sim \mathbf{m} + \sigma N(\mathbf{0}, \mathbf{C})$$

Example: Gaussian search distribution

The basic equation: sampling

$$\mathbf{x}_i^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} N(\mathbf{0}, \mathbf{C}^{(g)}) \quad \text{for } i = 1, \ldots, \lambda$$

$\mathbf{x}_i^{(g+1)} \in$ ¡ ^{n} is the i-th sampled search point

$\sim$ dontes the same distribution

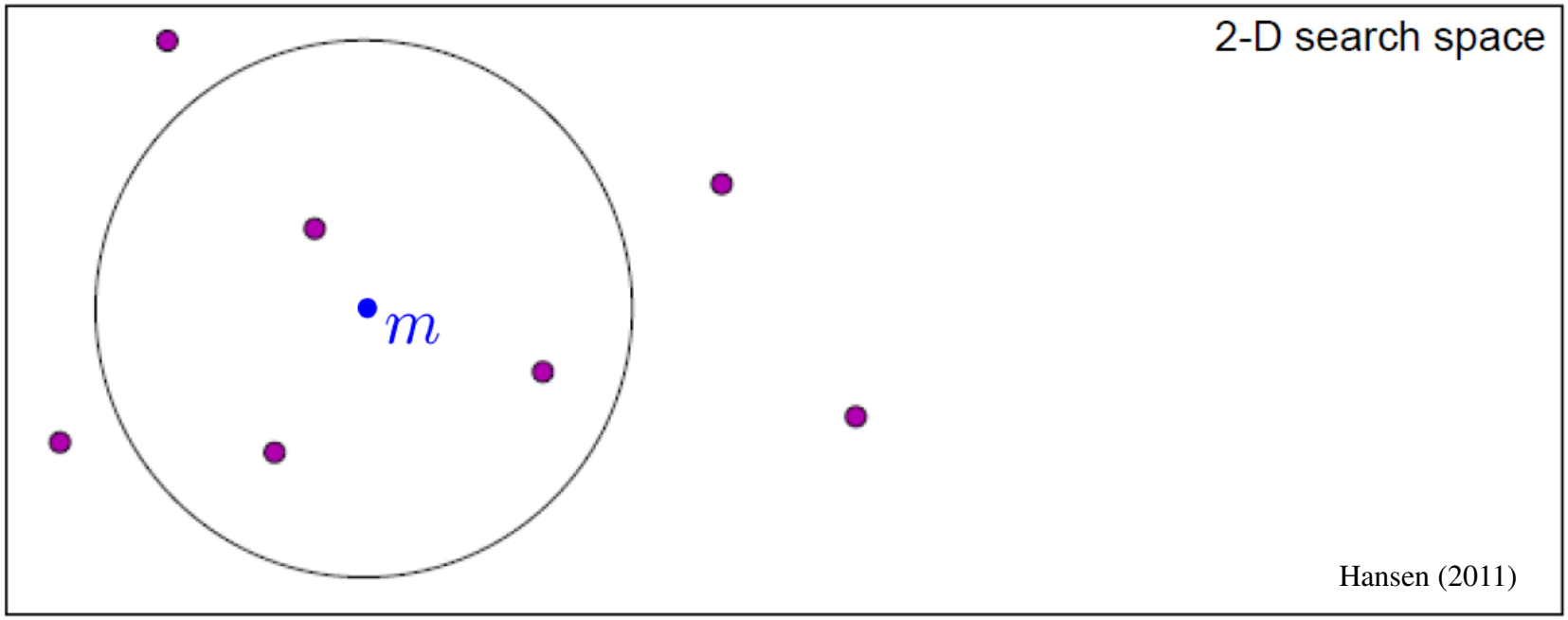
$\mathbf{m}^{(g)} \in$ ¡ ^{n} is the mean value of the search distribution

$\sigma^{(g)} \in$ ¡ $_+$ is the "overall" standard deviation, step size

$N(\mathbf{0}, \mathbf{C}^{(g)})$ is a multivariate normal distribution with zero mean and covariance matrix $\mathbf{C}^{(g)} \in$ ¡ ^{nxn}

$\lambda \geq 2$ is the population size, generation size, iteration size, sample size, number of solutions, etc.

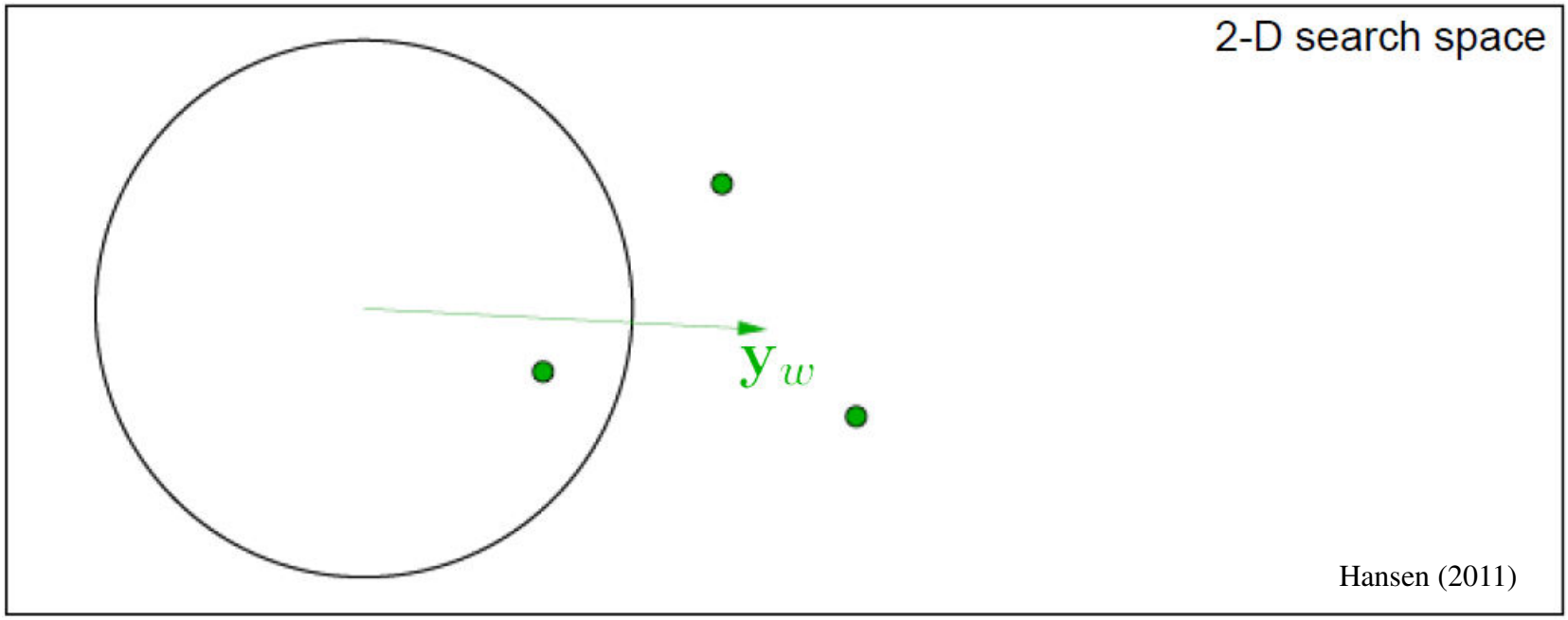
Example: Gaussian search distribution



sample from the initial distribution: $\mathcal{N}(m, \mathbf{I})$, $\mathbf{C} = \mathbf{I}$

$$\mathbf{x}_i = m + \sigma \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

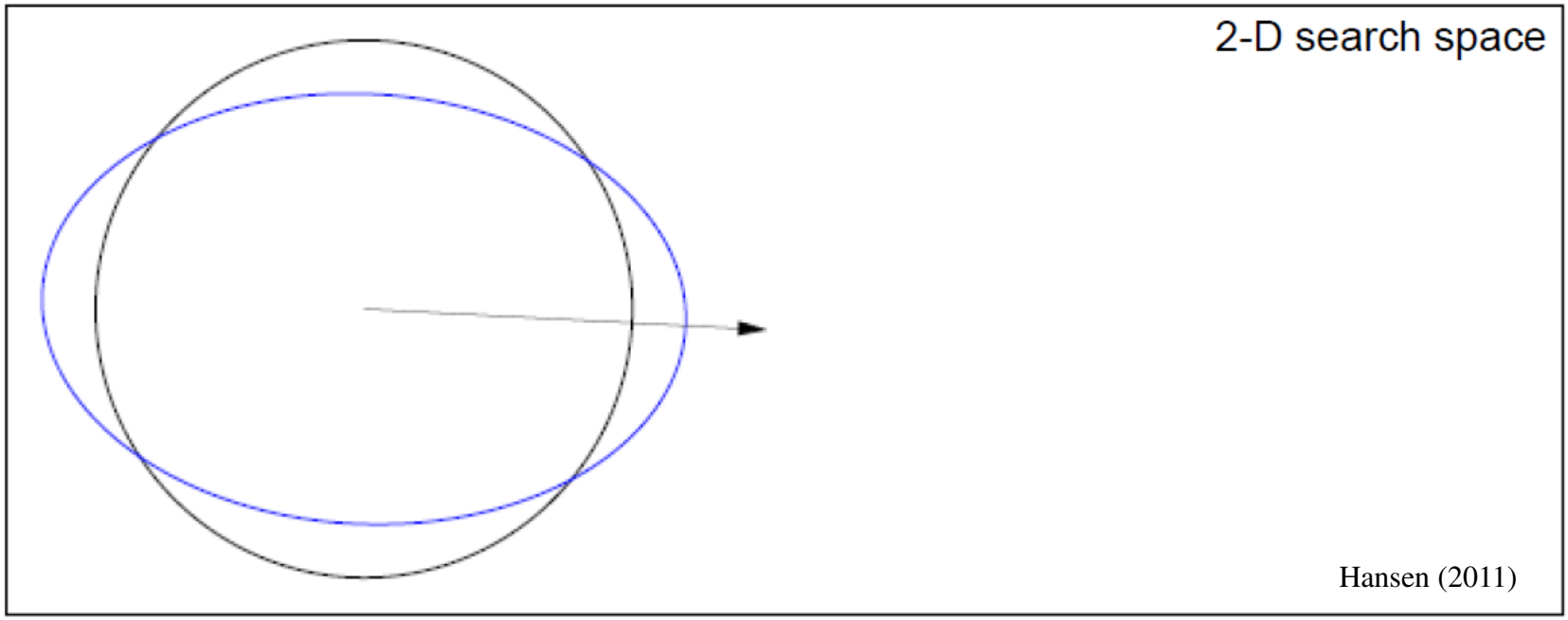
Example: Gaussian search distribution



$\mathbf{y}_w$ is the move of the mean m, disregarding σ

$$\mathbf{x}_i = m + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

Example: Gaussian search distribution

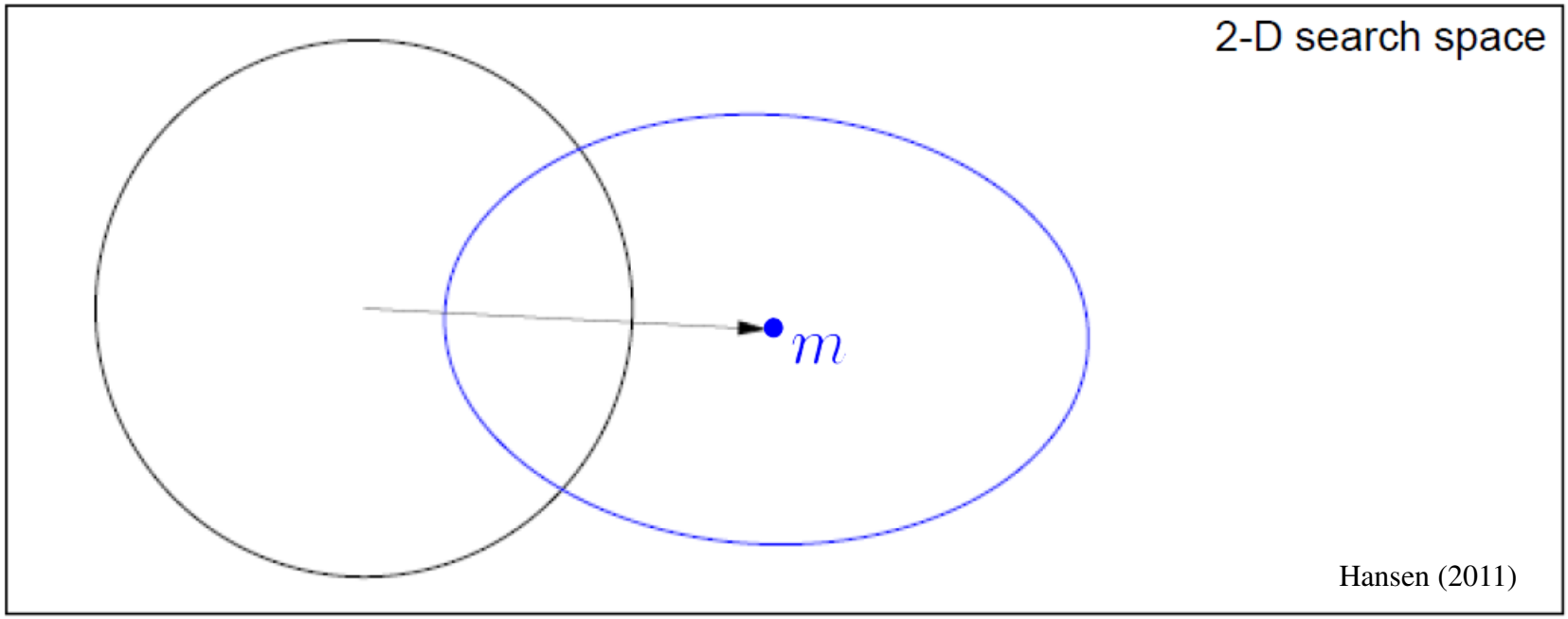


mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

$$\mathbf{C} \leftarrow 0.8\,\mathbf{C} + 0.2\,\mathbf{y}_w\mathbf{y}_w^{\mathrm{T}}$$

$$\mathbf{x}_i = m + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

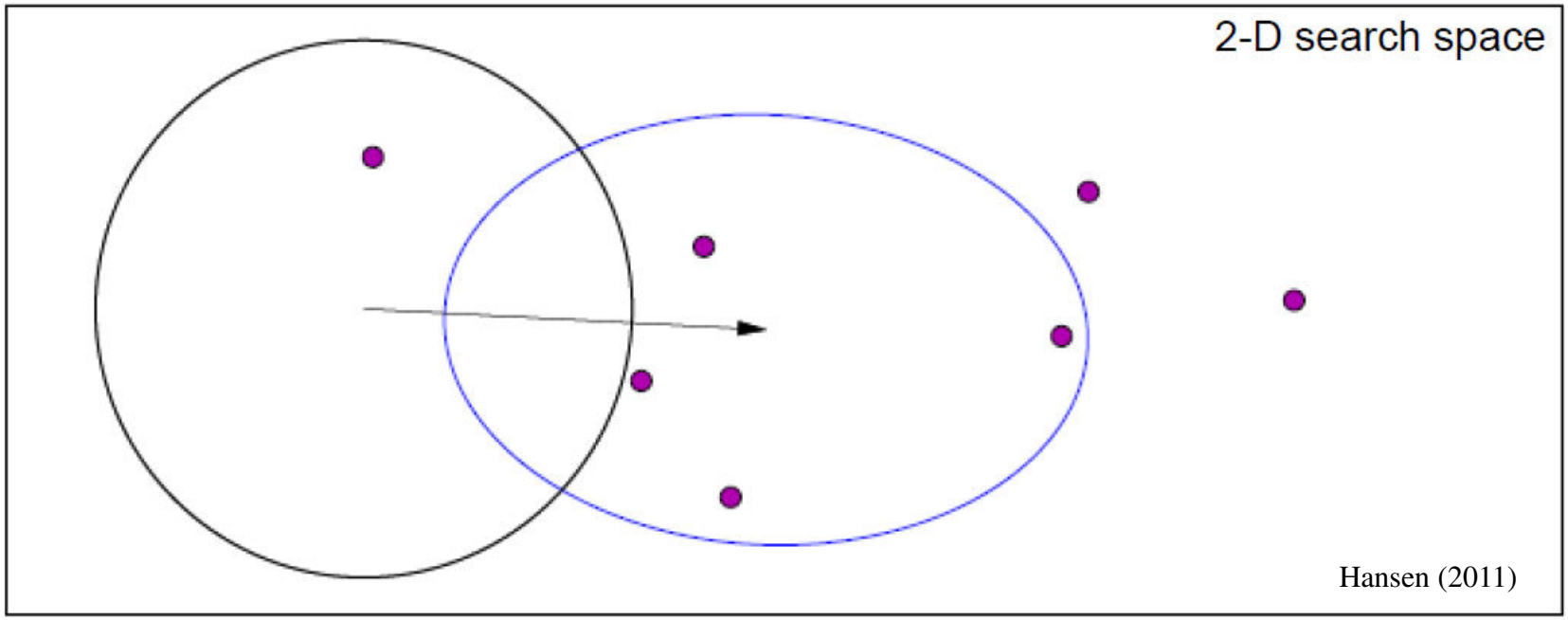
Example: Gaussian search distribution



new distribution $\mathcal{N}(m, \mathbf{C})$ (disregarding σ)

$$\mathbf{x}_i = m + \sigma\, \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

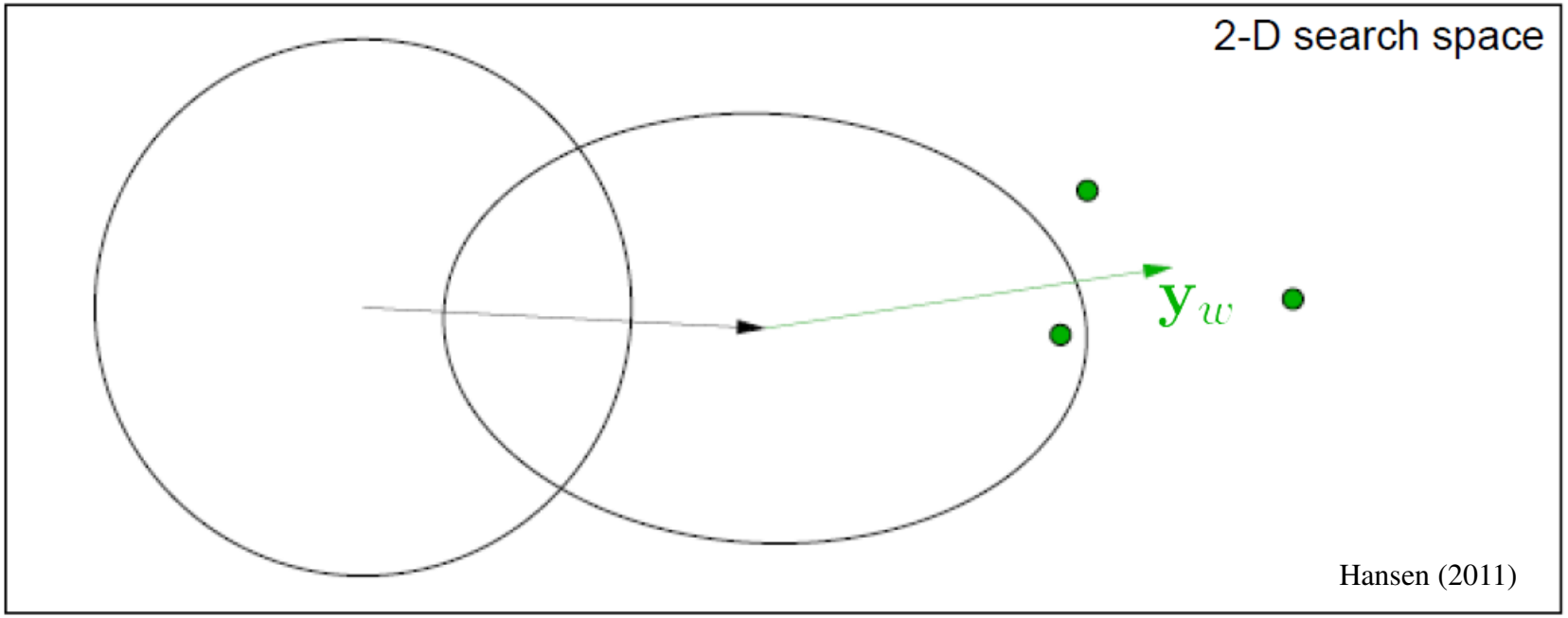
Example: Gaussian search distribution



another sample $\mathbf{x}_i \sim \mathcal{N}_i(m, \mathbf{C})$

$$\mathbf{x}_i = m + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

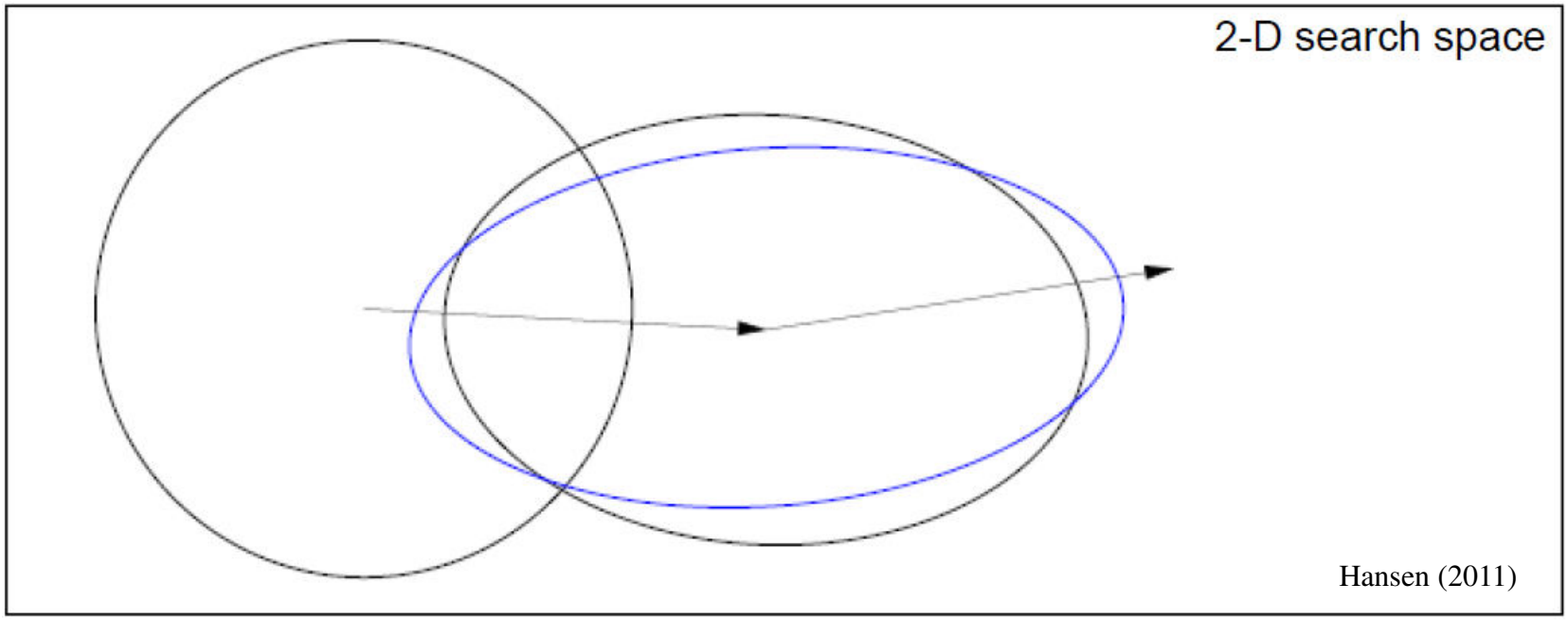
Example: Gaussian search distribution



$\mathbf{y}_w$, movement of the population mean m

$$\mathbf{x}_i = m + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

Example: Gaussian search distribution

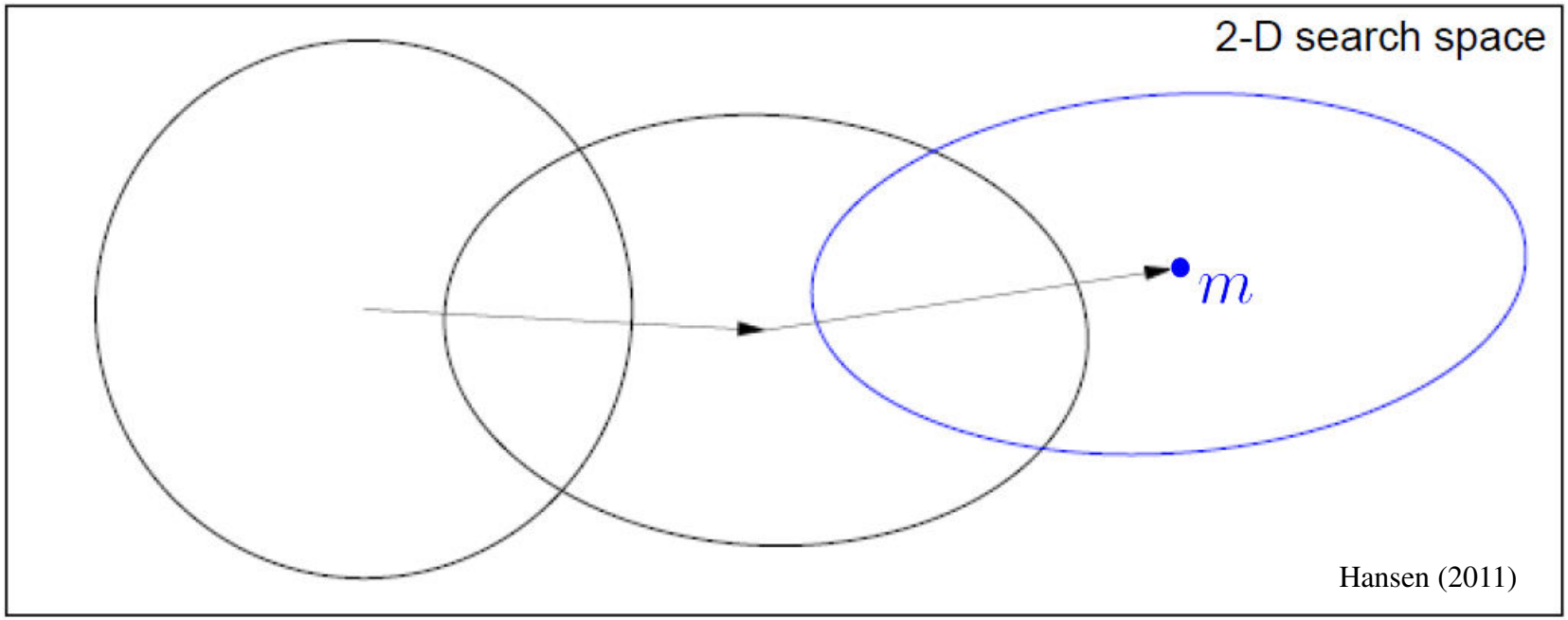


mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

$$\mathbf{C} \leftarrow 0.8\,\mathbf{C} + 0.2\,\mathbf{y}_w\mathbf{y}_w^{\mathrm{T}}$$

$$\mathbf{x}_i = m + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

Example: Gaussian search distribution



new distribution $\mathcal{N}(m, \mathbf{C}) = m + \mathcal{N}(\mathbf{0}, \mathbf{C})$

$$\mathbf{x}_i = m + \sigma\, \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

Covariance Matrix Adaption (CMA-ES)

From 1990s : Covariance Matrix Adaptation

$$\theta = (\mathbf{m}, \sigma, \mathbf{C}, p_\sigma, p_c), \quad p_\theta(\mathbf{x}) = N(\mathbf{x} \mid \mathbf{m}, \sigma^2 \mathbf{C})$$

where C is the covariance matrix of the search distribution

The θ maintains two more pieces of information:

p_σ and p_c capture the "path"(motion) of the mean $\mathbf{m}$ in recent iterations

Rough outline of the θ update:

- Let D'= best μ of D be the set of selected points
- Compute the new mean $\mathbf{m}$ from D'
- Update p_σ and p_c proportional to $\mathbf{m}_{k+1} - \mathbf{m}_k$
- Update σ depending on $|p_c|$
- Update C depending on $p_c p_c^T$ (rank-1-update) and Var(D') (rank-μ-update)

Yet before we get into CMA-ES:
We need to understand some general concepts

Some general concepts to understand

- The multivariate distribution
- Selection and recombination $\boldsymbol{m}$
- Step size control $\boldsymbol{\sigma}$
- The rank-1 update $\mathbf{C}$
- The rank-μ update $\mathbf{C}$

The multivariate normal distribution

Why Normal Distributions?

(1) Widely observed in nature

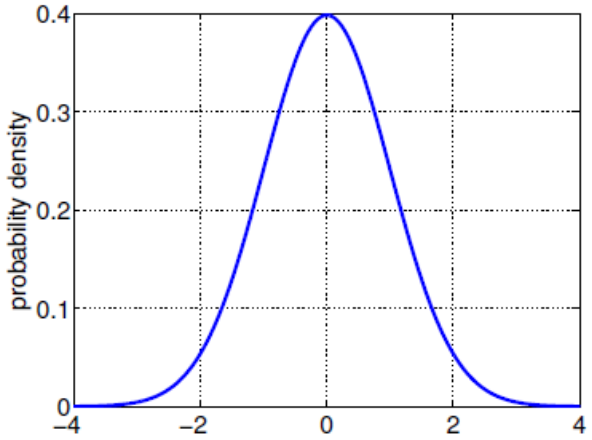
(2) Only stable distribution with finite variance (stable means that the sum of normal variates is again normal)

$$N(x, A) + N(y, B) \sim N(x + y, A + B)$$

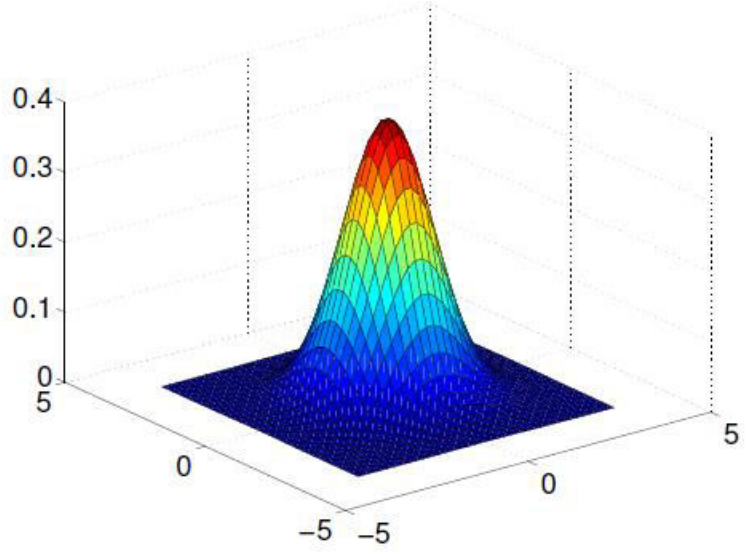
(3) Most convenient way to generate isotropic search points the isotropic distribution does not favor any direction, rotational invariant

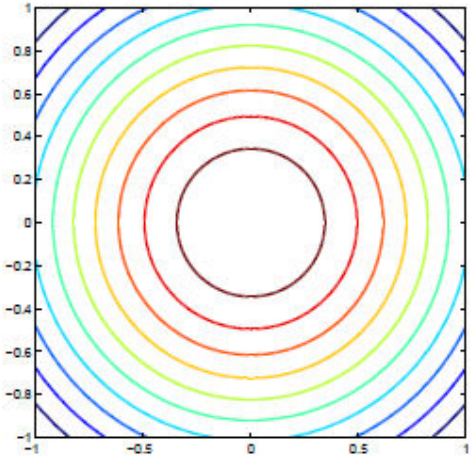
(4) Maximum entropy distribution with finite variance the least possible assumptions on f in the distribution shape

The multivariate normal distribution



Probability density of a 1-D normal distribution



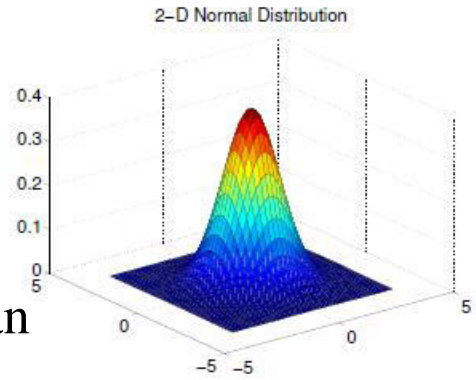


Probability density of a 2-D normal distribution

Auger and Hansen (2013)

The multivariate normal distribution

Any multivariate normal distribution $N(m, C)$ is uniquely determined by its mean value $m \in ¡^{n}$ and its symmetric positive definite $n \times n$ covariance matrix C.



The mean value m

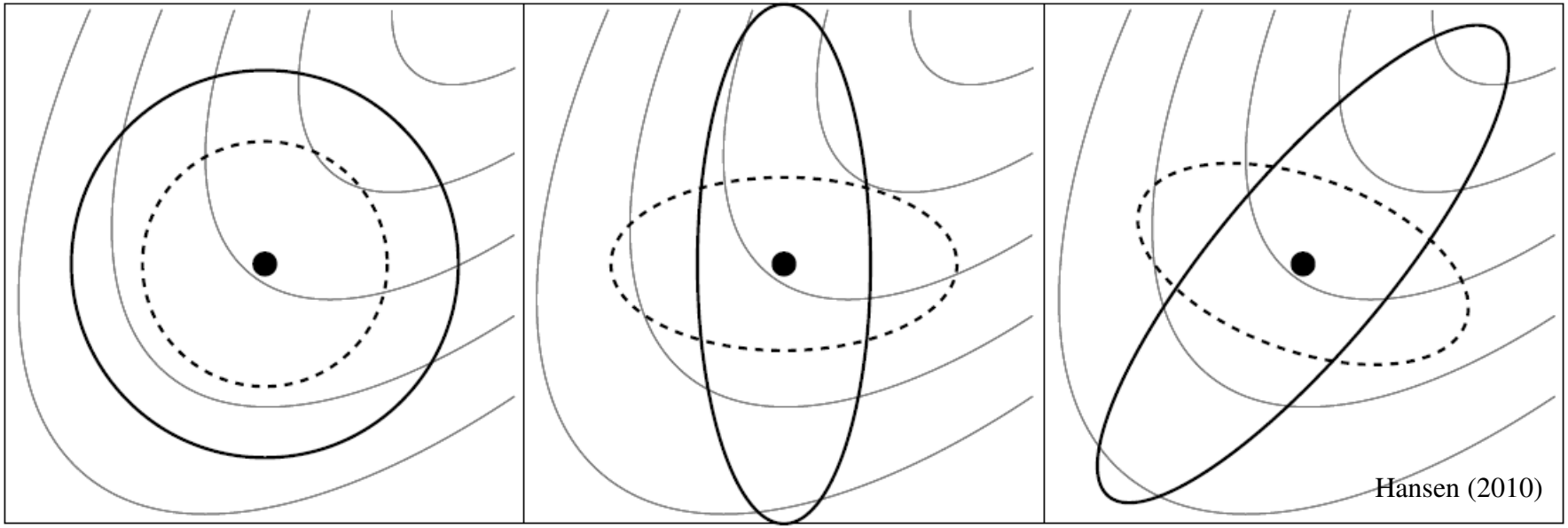
- determines the displacement (translation)
- value with the largest density (modal value)
- the distribution is symmetric about the distribution mean

The covariance matrix C

- determines the shape
- geometrical interpretation: iso-density ellipsoid

The multivariate normal distribution

Equal density all directions | Diagonal matrix only | Full covariance matrix



$N(\mathbf{m}, \sigma^2\mathbf{I}) \sim m + \sigma N(\mathbf{0}, \mathbf{I})$

One degree of freedom σ

components are independent standard normally distributed

$N(\mathbf{m}, \mathbf{D}^2) \sim \mathbf{m} + \mathbf{D}N(\mathbf{0}, \mathbf{I})$

***n* degrees of freedom**

components are independent, scaled

$N(\mathbf{m}, \mathbf{C}) \sim m + \mathbf{C}^{1/2}N(\mathbf{0}, \mathbf{I})$

$(n^2 + n)/2$ degrees of freedom

components are correlated

$N(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + N(\mathbf{0}, \mathbf{C})$

$\sim \mathbf{m} + \mathbf{C}^{1/2}N(\mathbf{0}, \mathbf{I})$

$\sim \mathbf{m} + \mathbf{B}\mathbf{D}\mathbf{B}^T\underbrace{N(\mathbf{0}, \mathbf{I})}_{\sim N(\mathbf{0}, \mathbf{I})}$

$\sim \mathbf{m} + \mathbf{B}\underbrace{\mathbf{D}N(\mathbf{0}, \mathbf{I})}_{\sim N(\mathbf{0}, \mathbf{D}^2)}$

$N(0, I)$ produces a spherical (istropic) distribution

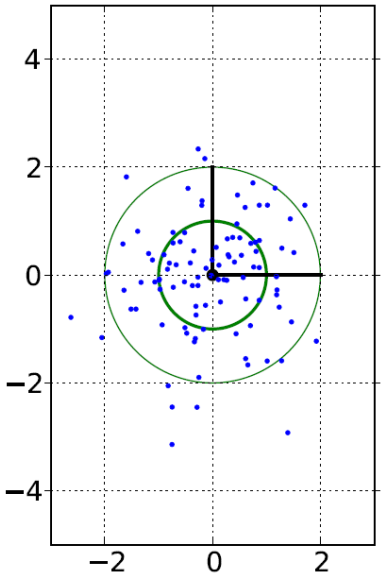
D scales the spherical distribution within the coordinate axes

B define new orinatation for the ellipsoid

$N(0, I)$ is a vector of (0,1)-normally distributed numbers

The multivariate normal distribution

Equal density all directions

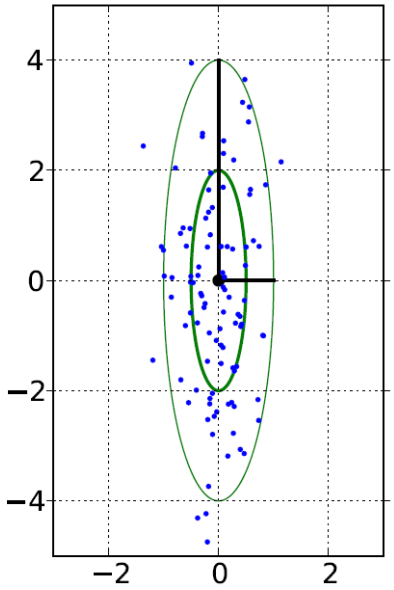


$N(\mathbf{m}, \sigma^2\mathbf{I}) \sim m + \sigma N(\mathbf{0}, \mathbf{I})$

One degree of freedom σ

components are independent standard normally distributed

Diagonal matrix only

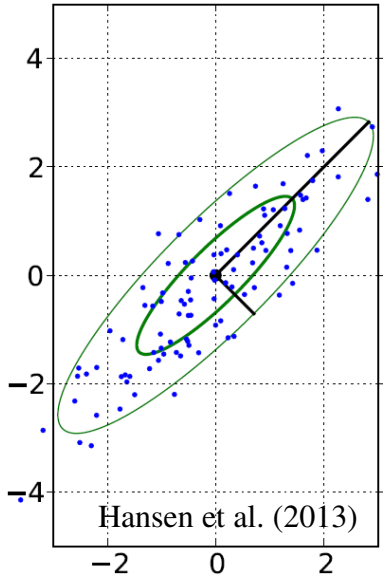


$N(\mathbf{m}, \mathbf{D}^2) \sim \mathbf{m} + \mathbf{D}N(\mathbf{0}, \mathbf{I})$

***n* degrees of freedom**

components are independent, scaled

Full covariance matrix



$N(\mathbf{m}, \mathbf{C}) \sim m + \mathbf{C}^{1/2}N(\mathbf{0}, \mathbf{I})$

$(\mathbf{n}^2 + \mathbf{n})/2$ degrees of freedom

components are correlated

$N(\mathbf{m}, \mathbf{C}) \sim \mathbf{m} + N(\mathbf{0}, \mathbf{C})$

$\sim \mathbf{m} + \mathbf{C}^{1/2}N(\mathbf{0}, \mathbf{I})$

$\sim \mathbf{m} + \mathbf{B}\mathbf{D}\mathbf{B}^T\underbrace{N(\mathbf{0}, \mathbf{I})}_{\sim N(\mathbf{0}, \mathbf{I})}$

$\sim \mathbf{m} + \mathbf{B}\underbrace{\mathbf{D}N(\mathbf{0}, \mathbf{I})}_{\sim N(\mathbf{0}, \mathbf{D}^2)}$

$N(0, I)$ produces a spherical (istropic) distribution

D scales the spherical distribution within the coordinate axes

B define new orinatation for the ellipsoid

$N(0, I)$ is a vector of (0,1)-normally distributed numbers

Selection and recombination

The best $\mu < \lambda$ search points are selected and weighted intermediate recombination is applied.

Given the i-th search point $\mathbf{x}_i = \mathbf{m} + \sigma \underbrace{N_i(\mathbf{0}, \mathbf{C})}_{=:\mathbf{y}_i} = \mathbf{m} + \sigma \mathbf{y}_i$

Let $x_{i:\lambda}$ the i-th ranked search point, such that $f(x_{i:\lambda}) \leq \ldots \leq f(x_{\lambda:\lambda})$

The new mean reads

$$\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \underbrace{\sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}}_{=:\mathbf{y}_w}$$

where

$$w_1 \geq \ldots \geq w_\mu > 0, \qquad \sum_{i=1}^{\mu} w_i = 1$$

Note:

$$\mathbf{y}_{i:\lambda}^{(g+1)} = (x_{i:\lambda}^{(g+1)} - m^{(g)}) / \sigma^{(g)}$$

The measure μ_w is the varaince effective selection mass

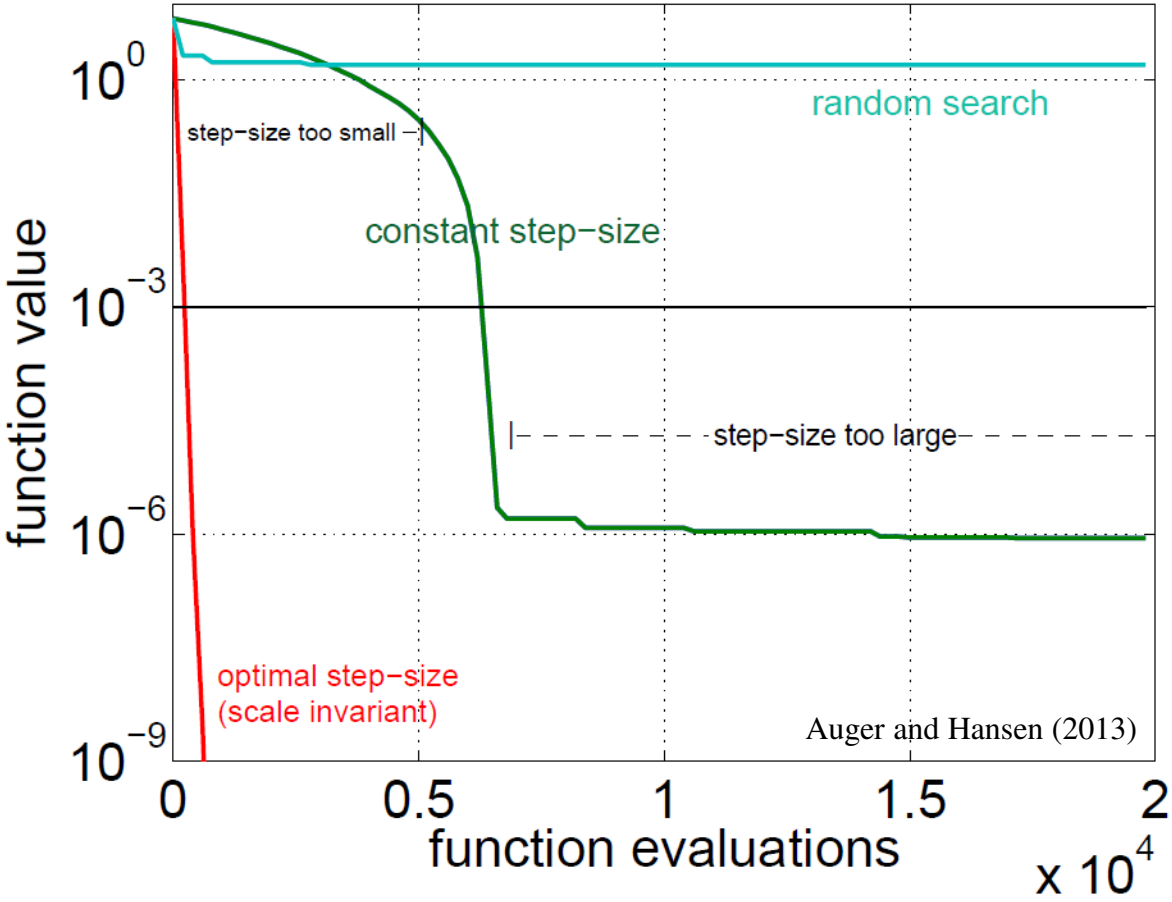
From $\sum_{i=1}^{\mu} w_i = 1$ we get $1 \leq \mu_w \leq \mu$ and $\mu_w = \mu$ for equal recommindation weights

$\mu_w = \lambda / 4$ indicates a reasonable setting of w_i

Step size control

Why step size control?

- The largest reliable learning rate for the covariance matrix update is too slow to achieve competitive change rates for the overall step length
- Step size controls the "overall scale" of the distribution



$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-2.2, 0.8]^n$
for $n = 10$

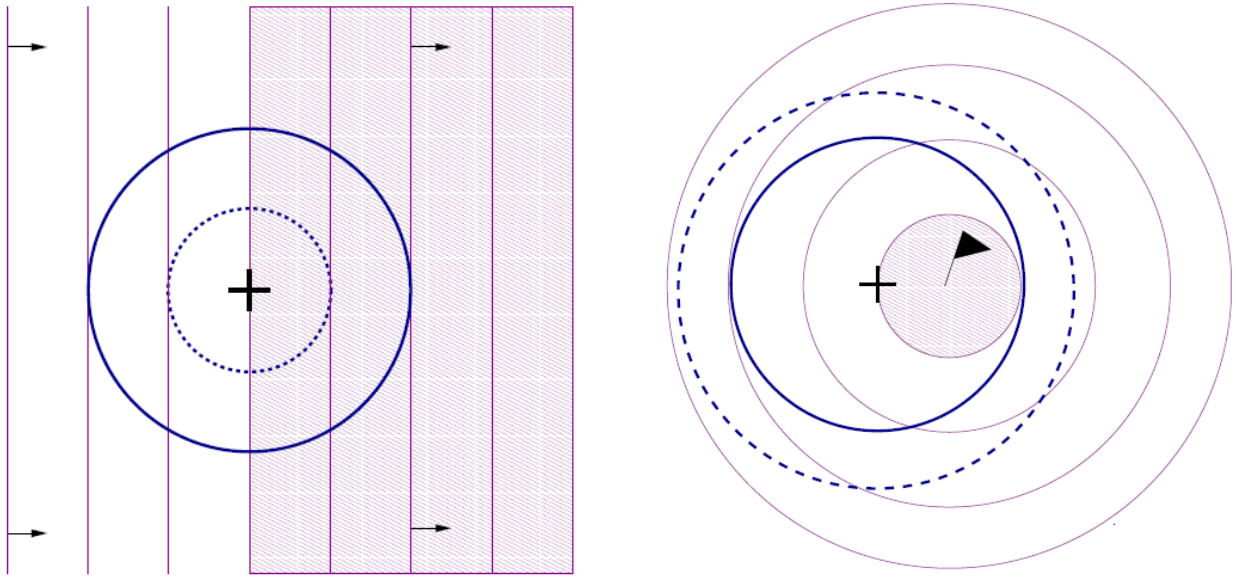
Step size control

<u>Method 1:</u> 1/5-th success rule
Increase step-size if more than 20% of the new solutions are successful, decrease otherwise

Algorithm 1 The (1+1)-ES with 1/5th Rule

0 **given** $n \in \mathbb{N}_+$, $d \approx \sqrt{n+1}$
1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\sigma > 0$
2 **while** not happy
3 $\quad \boldsymbol{x}_1 = \boldsymbol{x} + \sigma \times \mathcal{N}(\mathbf{0}, \mathbf{I})$ // mutation
4 $\quad \sigma \leftarrow \sigma \times \exp^{1/d}(\mathbb{1}_{f(\boldsymbol{x}_1) \leq f(\boldsymbol{x})} - 1/5)$
5 $\quad$ **if** $f(\boldsymbol{x}_1) \leq f(\boldsymbol{x})$ // select if better
6 $\quad\quad \boldsymbol{x} = \boldsymbol{x}_1$ // $\boldsymbol{x}$-value of new parent

Hansen et al. (2013)



Auger and Hansen (2013)

Step size control

<u>Method 2:</u> σ-self-adaptation
Mutation is applied to the step-size and the better (according to the objective function value) is selected

Algorithm 2 The $(\mu/\mu, \lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, $\lambda \geq 5n$, $\mu \approx \lambda/4 \in \mathbb{N}$, $\tau \approx 1/\sqrt{n}$, $\tau_\mathrm{i} \approx 1/n^{1/4}$
1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$
2 **while** not happy
3 **for** $k \in \{1, \ldots, \lambda\}$
// random numbers i.i.d. for all k
4 $\xi_k = \tau \mathcal{N}(0,1)$ // global step-size
5 $\boldsymbol{\xi}_k = \tau_\mathrm{i} \mathcal{N}(\mathbf{0}, \mathbf{I})$ // coordinate-wise $\boldsymbol{\sigma}$
6 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ //$\boldsymbol{x}$-vector change
// mutation
7 $\boldsymbol{\sigma}_k = \boldsymbol{\sigma} \circ \exp(\boldsymbol{\xi}_k) \times \exp(\xi_k)$
8 $\boldsymbol{x}_k = \boldsymbol{x} + \boldsymbol{\sigma}_k \circ \boldsymbol{z}_k$
9 $\mathcal{P} = \mathsf{sel_}\mu\mathsf{_best}(\{(\boldsymbol{x}_k, \boldsymbol{\sigma}_k, f(\boldsymbol{x}_k)) \,|\, 1 \leq k \leq \lambda\})$
// recombination
10 $\boldsymbol{\sigma} = \frac{1}{\mu} \sum_{\boldsymbol{\sigma}_k \in \mathcal{P}} \boldsymbol{\sigma}_k$
11 $\boldsymbol{x} = \frac{1}{\mu} \sum_{\boldsymbol{x}_k \in \mathcal{P}} \boldsymbol{x}_k$

Hansen et al. (2013)

Step size control

<u>Method 3:</u> Derandomized σ-self-adaptation

Mutation is applied similar to σ-self-adaptation yet without introducing new variation term for σ and using the variation of $z_k=(x_k-m)/\sigma$ with damping terms. This is to reduce bias due to stochastic fluctuations of σ.

Algorithm 3 Derandomized $(1,\lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, $\lambda \approx 10$, $\tau \approx 1/3$, $d \approx \sqrt{n}$, $d_\mathrm{i} \approx n$

1 **initialize** $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$

2 **while** not happy

3 **for** $k \in \{1, \ldots, \lambda\}$

// random numbers i.i.d. for all k

4 $\xi_k = \tau\,\mathcal{N}(0,1)$

5 $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$

// mutation, re-using random events

6 $\boldsymbol{x}_k = \boldsymbol{x} + \exp(\xi_k) \times \boldsymbol{\sigma} \circ \boldsymbol{z}_k$

7 $\boldsymbol{\sigma}_k = \boldsymbol{\sigma} \circ \exp^{1/d_\mathrm{i}}\left(\frac{|\boldsymbol{z}_k|}{\mathbb{E}|\mathcal{N}(0,1)|} - \mathbf{1}\right)$

7b $\times \exp^{1/d}(\xi_k)$

8 $(\boldsymbol{x}_1, \boldsymbol{\sigma}_1, f(\boldsymbol{x}_1)) \leftarrow$ select_single_best($\{(\boldsymbol{x}_k, \boldsymbol{\sigma}_k, f(\boldsymbol{x}_k)) \mid 1 \le k \le \lambda\}$)

// assign new parent

9 $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1$

10 $\boldsymbol{x} = \boldsymbol{x}_1$

Hansen et al. (2013)

Step size control

<u>Method 4:</u> Non-local Derandomized σ-self-adaptation

To avoid small increments, path length control (cumulative step-size adaptation) using the search (evolution) path $\mathbf{s}_\sigma$,which is an exponentially fading record of mutation steps. This is basically a sequence or sum of consecutive successful z-steps that is non-local in time and space. Thus, the search path carries information about the interrelation between successive steps.

Algorithm 4 The $(\mu/\mu, \lambda)$-ES with Search Path

given $n \in \mathbb{N}_+$, $\lambda \in \mathbb{N}$, $\mu \approx \lambda/4 \in \mathbb{N}$, $c_\sigma \approx \sqrt{\mu/(n+\mu)}$, $d \approx 1 + \sqrt{\mu/n}$, $d_\mathrm{i} \approx 3n$

initialize $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{\sigma} \in \mathbb{R}^n_+$, $\boldsymbol{s}_\sigma = \mathbf{0}$

while not happy

  for $k \in \{1, \ldots, \lambda\}$

    $\boldsymbol{z}_k = \mathcal{N}(\mathbf{0}, \mathbf{I})$ // i.i.d. for each k

    $\boldsymbol{x}_k = \boldsymbol{x} + \boldsymbol{\sigma} \circ \boldsymbol{z}_k$

  $\mathcal{P} \leftarrow \mathsf{sel_}\mu\mathsf{_best}(\{(\boldsymbol{x}_k, \boldsymbol{z}_k, f(\boldsymbol{x}_k)) \,|\, 1 \le k \le \lambda\})$

  // recombination and parent update

  $\boldsymbol{s}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{s}_\sigma +$

  $$\sqrt{c_\sigma\,(2 - c_\sigma)}\,\frac{\sqrt{\mu}}{\mu} \sum_{\boldsymbol{z}_k \in \mathcal{P}} \boldsymbol{z}_k$$

  $$\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} \circ \exp^{1/d_\mathrm{i}} \left(\frac{|\boldsymbol{s}_\sigma|}{\mathbb{E}|\mathcal{N}(0,1)|} - \mathbf{1} \right)$$

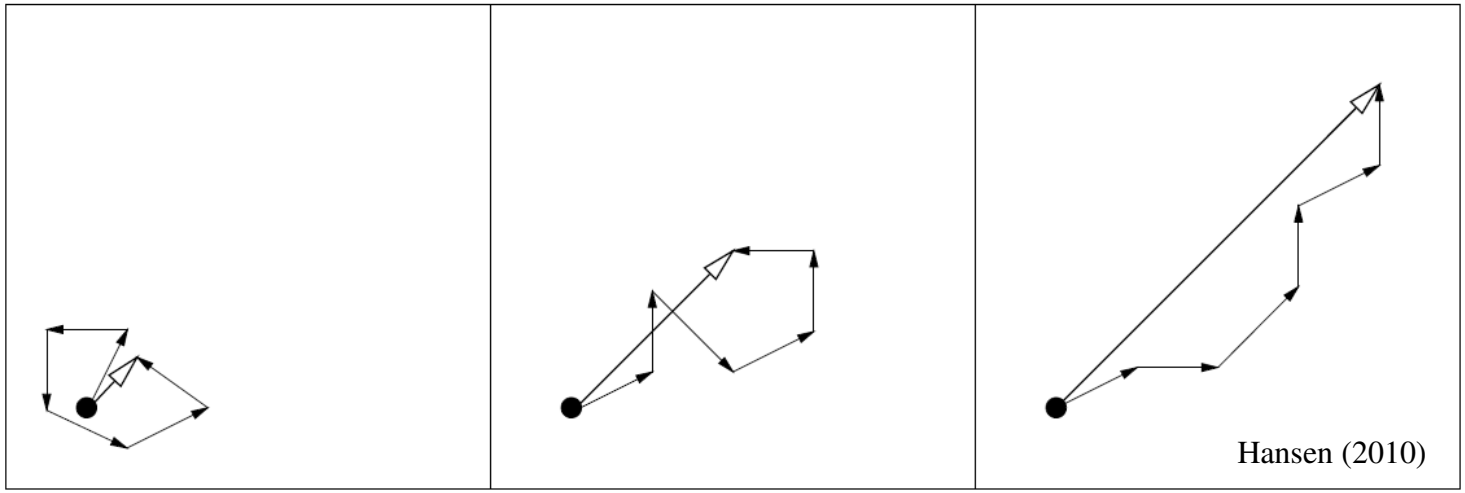
  $$\times \exp^{c_\sigma/d} \left(\frac{\|\boldsymbol{s}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right)$$

  $$\boldsymbol{x} = \frac{1}{\mu} \sum_{\boldsymbol{x}_k \in \mathcal{P}} \boldsymbol{x}_k$$

Hansen et al. (2013)

Step size control

Non-local Derandomized σ-self-adaptation: cumulative step-size adaptation that measures the length of the search (evolution) path: the pathway of the mean vector ***m*** in the generation sequence



Decrease σ

Increase σ

Rank-μ update

$$\mathbf{x}_i \leftarrow \mathbf{m} + \sigma \mathbf{y}_i, \qquad \mathbf{y}_i \sim N_i(\mathbf{0}, \mathbf{C})$$

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w, \qquad \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$$

The rank-μ update extends the update rule for large population sizes λ using μ vectors to update **C** at each generation step.

The weighted empirical covariance matrix

$$\mathbf{C}_\mu = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^{\mathrm{T}}$$

Computes a weighted mean of the outer products of the best μ steps

They rank-μ reads

$$\mathbf{C} \leftarrow (1 - c_{\mathrm{cov}})\mathbf{C} + c_{\mathrm{cov}} C_\mu$$

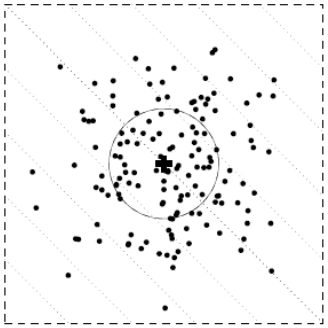
where $c_{\mathrm{cov}} \approx \mu_w / n^2$ and $c_{\mathrm{cov}} \leq 1$

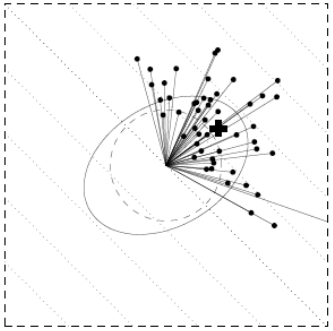
Rank-μ update

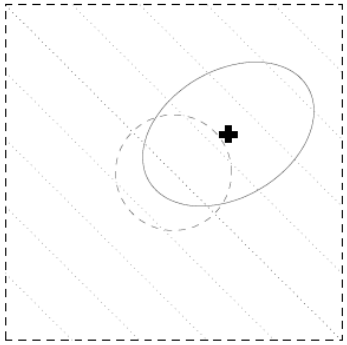
Sampling
λ= 150 solutions

Calculating **C** from
μ= 50 solutions

New distribution

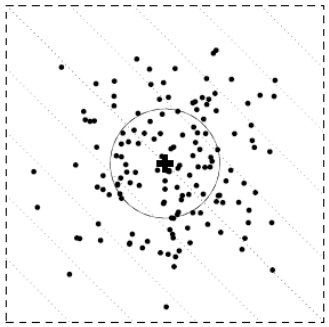


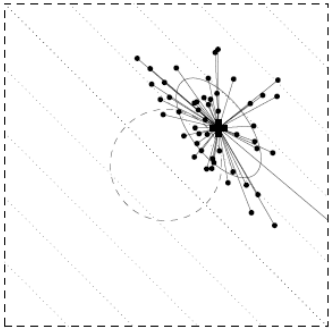


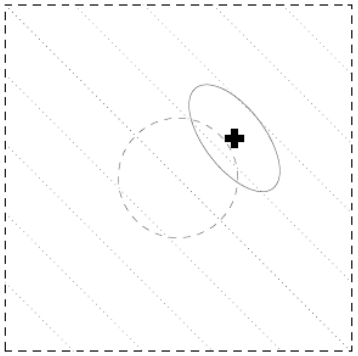


$$x_i = m_{\text{old}} + y_i, \quad y_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \qquad \mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{old}})(x_{i:\lambda} - m_{\text{old}})^{\mathrm{T}} \qquad m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum y_{i:\lambda}$$

Rank-μ CMA conducts a PCA of steps







$$x_i = m_{\text{old}} + y_i, \quad y_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \qquad \mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{new}})(x_{i:\lambda} - m_{\text{new}})^{\mathrm{T}} \qquad m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum y_{i:\lambda}$$

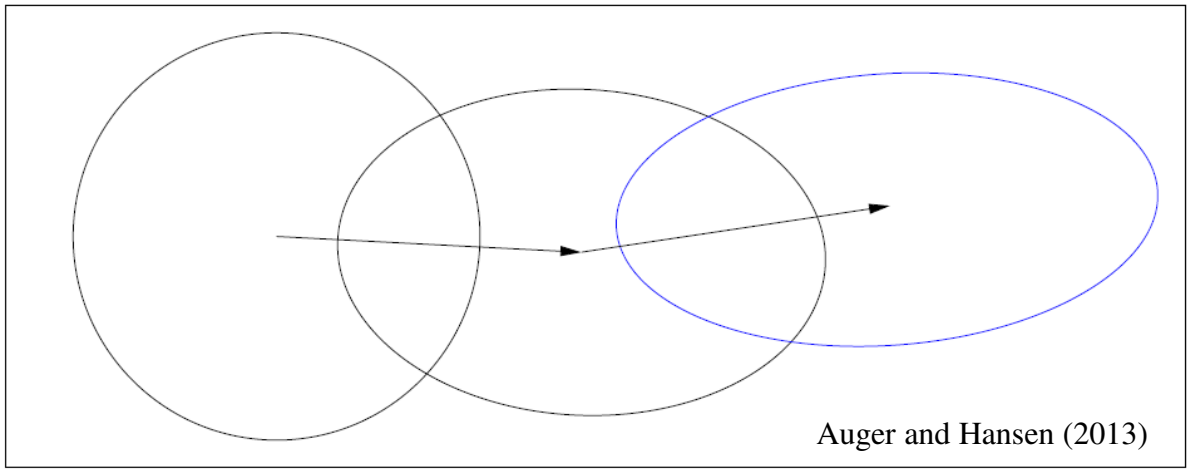
Estimation of Multivariate Normal Algorithm conducts a PCA of points

m_{new} is the minimizer for the variances when calculating C

Auger and Hansen (2013)

Rank-μ update

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w, \qquad \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}, \qquad \mathbf{y}_i \sim N_i(\mathbf{0}, \mathbf{C})$$



New distribution

$$C \leftarrow 0.8 \times C + 0.2 \times \mathbf{y}_w \mathbf{y}_w^T$$

$$C \leftarrow (1 - c_{\text{cov}}) C + c_{\text{cov}} \mu_w \mathbf{y}_w \mathbf{y}_w^T$$

Adaptation increases the likelihood of

successful points by updating $m \leftarrow m + y_w$

successful steps by updating $C \leftarrow 0.8\ C\ +\ 0.2\ y_w y_w^T$

Natural gradient desecent in m and C

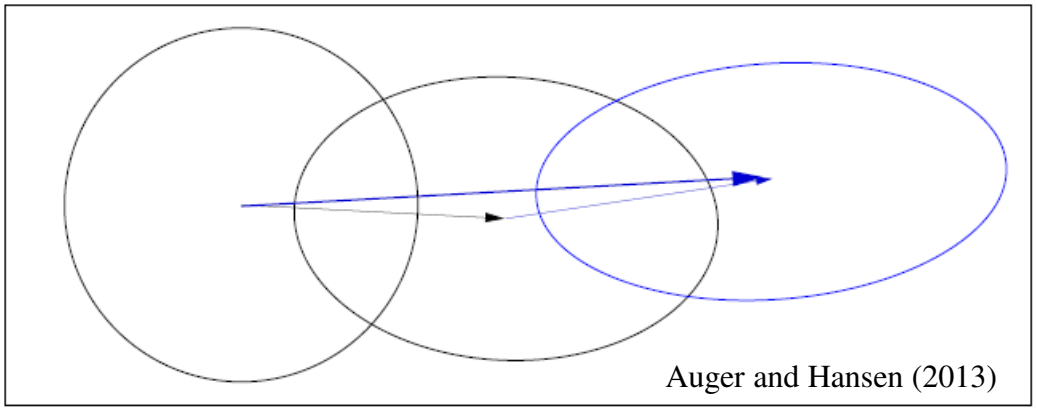
Rank-one update

In Rank-μ update we estimated the covariance matrix using all selected steps from a single generation.

We can also take the opposite viewpoint, which is to repeatedly update the convince matrix in generation sequence using a single selected step only!

Rank-one update

Evolution path: the search path the strategy takes over a number of generation steps. It can be expressed as a sum of consecutive steps of the mean ***m***.



An exponentially weighted sum of steps y_w is used

$$p_c \propto \sum_{i=0}^{g} \underbrace{(1-c_c)^{g-1}}_{\text{Exponentially fading weights}} \mathbf{y}_w^{(i)}$$

The recursive construction of the evolution path (cumulation)

$$p_c \leftarrow \underbrace{(1-c_c)}_{\text{Decay factor}} p_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu_w}}_{\text{Normalization factor}} \underbrace{\mathbf{y}_w}_{\text{Input}=\frac{\mathbf{m}-\mathbf{m}_{old}}{\sigma}}$$

Where $\mu_w = \dfrac{1}{\sum w_i^2}$, $\quad c_c \le 1$

History information is accumulated in the evolution path

Evolution path "Cumulation" is a widely used technique. Also know as:

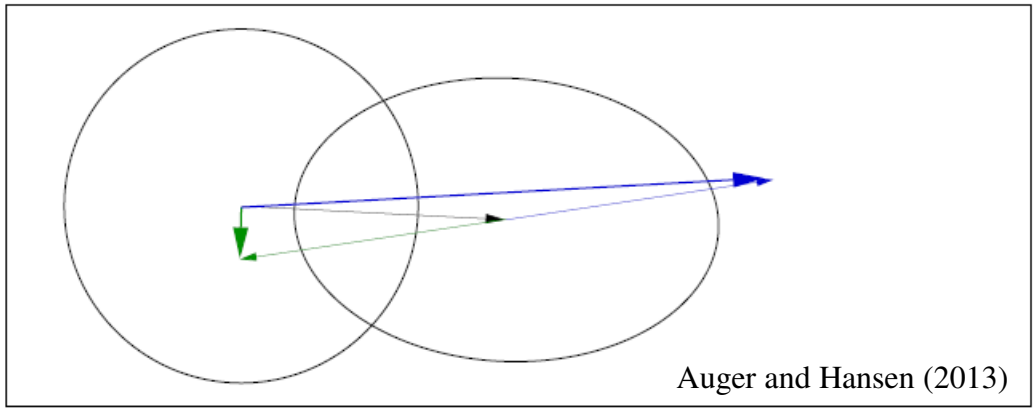
- Exponential smoothing in time series, forecasting
- Exponentially weighted moving average
- Iterate averaging in stochastic approximation

Rank-one update

Utilizing the Evolution path "Cumulation"

We used $\mathbf{y}_w \mathbf{y}_w^T$ for updating $\mathbf{C}$

With $\mathbf{y}_w \mathbf{y}_w^T = -\mathbf{y}_w (-\mathbf{y}_w)^T$ the sign of $\mathbf{y}_w$ is lost



Auger and Hansen (2013)

The sign information (signifying correlation between steps) is re-introduced by using the evolution path (a non-local trajectory)

$$p_c \leftarrow \underbrace{(1-c_c)}_{\text{Decay factor}}\, p_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu_w}}_{\text{Normalization factor}}\, \underbrace{\mathbf{y}_w}_{\text{Input}=\frac{\mathbf{m}-\mathbf{m}_{old}}{\sigma}}$$

$$\mathbf{C} \leftarrow (1-c_{\text{cov}})\mathbf{C} + c_{\text{cov}} \underbrace{\mathbf{p}_c \mathbf{p}_c^T}_{\text{Rank-one update}}$$

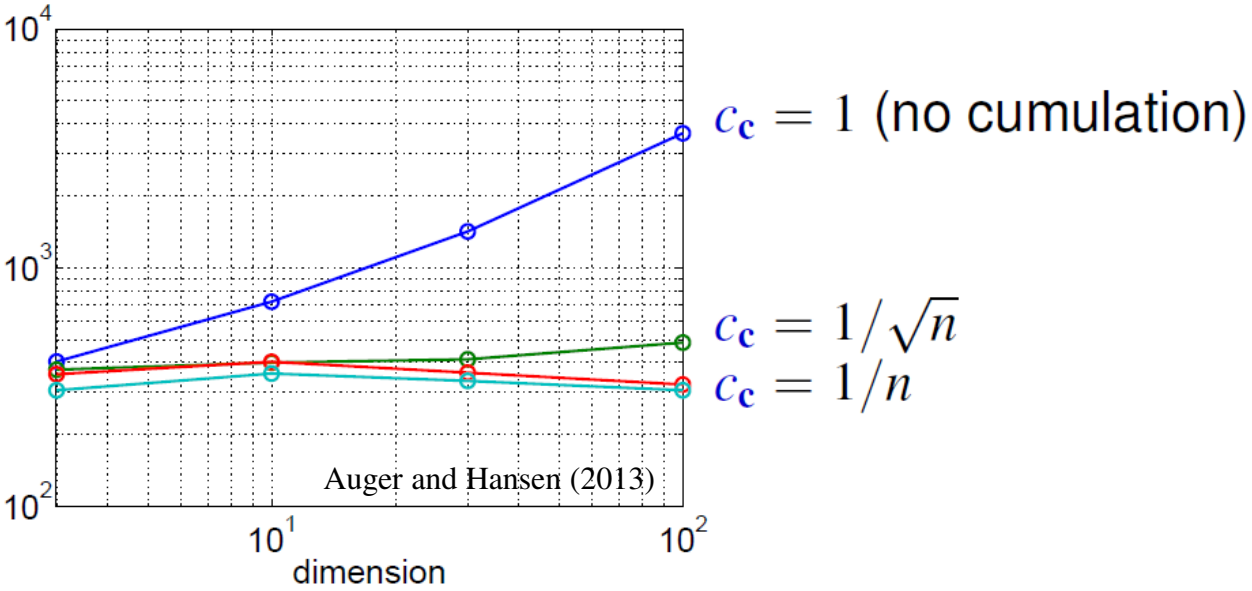
Where $\mu_w = \frac{1}{\sum w_i^2}$, $c_{\text{cov}} \le c_c \le 1$ such that $1/c_c$ is the "backward time horizon"

Rank-one update

Using an **evolution path** for the **rank-one update** of the covariance matrix reduces the number of function evaluations to adapt **from about** $O(n^2)$ **to** $O(n)$

Number of *f*-evaluations divided by dimension on the cigar function

$$f(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$$



The overall model complexity is n^2 but important parts of the model can be learned in time of order n

Combining rank-μ and rank-one update

The rank-μ update
Is the primary mechanism whenever a large population is used. Information within the population of one generation is used efficiently to increases the possible learning rate.

The rank-one update
Is in particular important in small populations. Information of correlations between generations is exploited by using the evolution path to reduce the number of necessary function evaluations.

Rank-one update and rank-μ update can be combined

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\mathbf{C} + c_1 \underbrace{\mathbf{p}_c \mathbf{p}_c^{\mathrm{T}}}_{\text{Rank-one update}} + c\mu \underbrace{\sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^{\mathrm{T}}}_{\text{Rank-}\mu\text{ update}}$$

Covariance Matrix Adaption

Covariance matrix adaptation

- Learns all pairwise dependencies between variables: off-diagonal entries in the covariance matrix reflect the dependencies
- Conducts a principle component analysis (PCA) of steps y_w, sequentially in time and space: eigenvectors of the covariance matrix C are the principle components / the principle axes of the mutation ellipsoid
- Learns a new rotated problem representation: components are independent (only) in the new representation

Covariance Matrix Adaption (CMA-ES)

From 1990s: Covariance Matrix Adaptation

$$\theta = (\mathbf{m}, \sigma, \mathbf{C}, p_\sigma, p_c), \quad p_\theta(\mathbf{x}) = N(\mathbf{x} \mid \mathbf{m}, \sigma^2 \mathbf{C})$$

where $\mathbf{C}$ is the covariance matrix of the search distribution

The θ maintains two more pieces of information:

p_σ and p_c capture the "path"(motion) of the mean $\mathbf{m}$ in recent iterations

Rough outline of the θ update:

- Let $D' =$ best μ of D be the set of selected points
- Compute the new mean $\mathbf{m}$ from D'
- Update p_σ and p_c proportional to $\mathbf{m}_{k+1} - \mathbf{m}_k$
- Update σ depending on $|p_\sigma|$
- Update $\mathbf{C}$ depending on $p_c p_c^T$ (rank-1-update) and Var(D') (rank-μ-update)

We can now get into CMA-ES:

5. CMA-ES : The Algorithm

CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$, and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$
While not terminate

$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\,\boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \quad \text{for } i = 1, \ldots, \lambda \qquad \text{sampling}$$

$$\boldsymbol{m} \leftarrow \textstyle\sum_{i=1}^{\mu} w_i\,\boldsymbol{x}_{i:\lambda} = \boldsymbol{m} + \sigma\boldsymbol{y}_w \quad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i\,\boldsymbol{y}_{i:\lambda} \qquad \text{update mean}$$

$$\boldsymbol{p}_\mathbf{c} \leftarrow (1 - c_\mathbf{c})\,\boldsymbol{p}_\mathbf{c} + \mathbb{1}_{\{\|\boldsymbol{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_\mathbf{c})^2}\sqrt{\mu_w}\,\boldsymbol{y}_w \qquad \text{cumulation for } \mathbf{C}$$

$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\,\boldsymbol{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2}\sqrt{\mu_w}\,\mathbf{C}^{-\frac{1}{2}}\boldsymbol{y}_w \qquad \text{cumulation for } \sigma$$

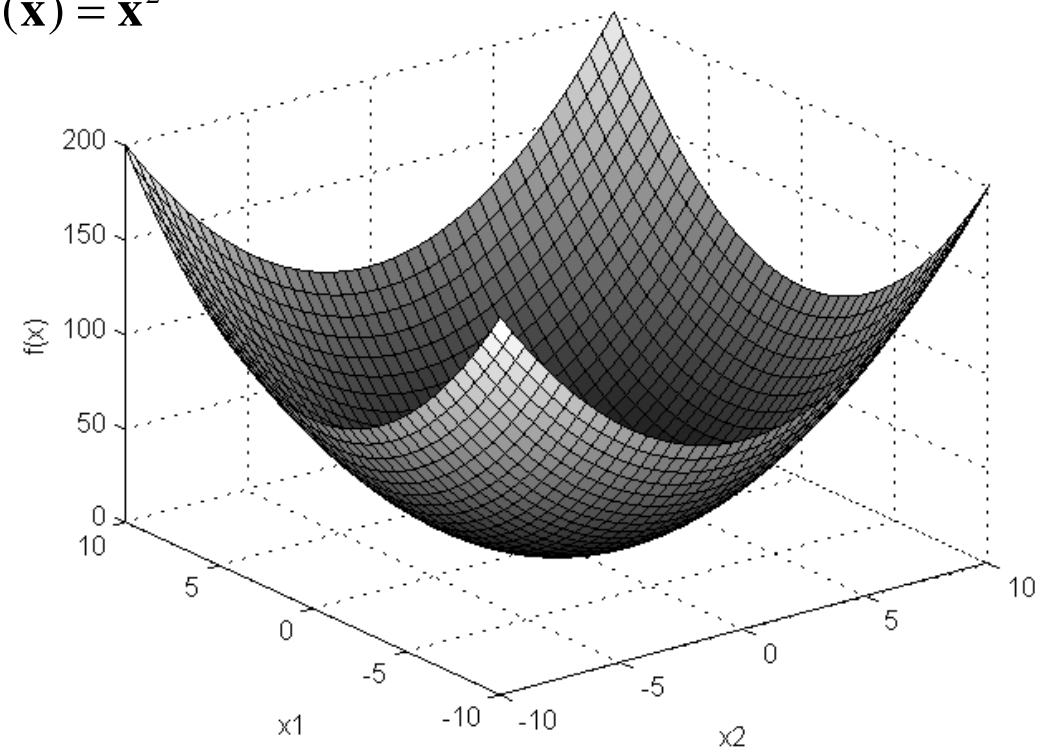
$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\,\mathbf{C} + c_1\,\boldsymbol{p}_\mathbf{c}\boldsymbol{p}_\mathbf{c}{}^{\mathrm{T}} + c_\mu \textstyle\sum_{i=1}^{\mu} w_i\,\boldsymbol{y}_{i:\lambda}\boldsymbol{y}_{i:\lambda}^{\mathrm{T}} \qquad \text{update } \mathbf{C}$$

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right) \qquad \text{update of } \sigma$$

Auger and Hansen (2013)

The simplest numerical example

$$f(\mathbf{x}) = \mathbf{x}^2$$



$$\text{find } \min_{\mathbf{x}} f(\mathbf{x}),$$

Internal parameters

- related to selection and recombination
  - λ, offspring number, new solutions sampled, population size
  - μ, parent number, solutions involved in updates of $\boldsymbol{m}$, $\mathbf{C}$, and σ
  - $w_{i=1,\ldots,\mu}$, recombination weights

    μ and w_i should be chosen such that the variance effective selection mass $\mu_w \approx \frac{\lambda}{4}$, where $\mu_w := 1/\sum_{i=1}^{\mu} {w_i}^2$.
- related to $\mathbf{C}$-update
  - $c_{\mathbf{c}}$, decay rate for the evolution path
  - c_1, learning rate for rank-one update of $\mathbf{C}$
  - c_μ, learning rate for rank-μ update of $\mathbf{C}$
- related to σ-update
  - c_σ, decay rate of the evolution path
  - d_σ, damping for σ-change

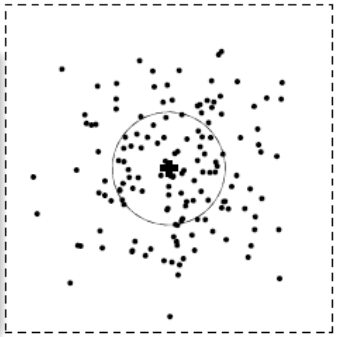
Auger and Hansen (2013)

Sampling

New search points are sampled normally distributed

$$\boldsymbol{x}_i \sim \boldsymbol{m} + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C}) \qquad \text{for } i = 1, \ldots, \lambda$$

as perturbations of $\boldsymbol{m}$, where $\boldsymbol{x}_i, \boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} \in \mathbb{R}^{n \times n}$



where

- the mean vector $\boldsymbol{m} \in \mathbb{R}^n$ represents the favorite solution
- the so-called step-size $\sigma \in \mathbb{R}_+$ controls the *step length*
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the **shape** of the distribution ellipsoid

here, all new points are sampled with the same parameters

The question remains how to update $\boldsymbol{m}$, $\mathbf{C}$, and σ.

Auger and Hansen (2013)

Mean update

Given the i-th solution point $\boldsymbol{x}_i = \boldsymbol{m} + \sigma \underbrace{\mathcal{N}_i(\mathbf{0}, \mathbf{C})}_{=: \boldsymbol{y}_i} = \boldsymbol{m} + \sigma \boldsymbol{y}_i$

Let $\boldsymbol{x}_{i:\lambda}$ the ***i*-th ranked** solution point, such that $f(\boldsymbol{x}_{1:\lambda}) \leq \cdots \leq f(\boldsymbol{x}_{\lambda:\lambda})$
The new mean reads

$$\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i \, \boldsymbol{x}_{i:\lambda} = \boldsymbol{m} + \sigma \underbrace{\sum_{i=1}^{\mu} w_i \, \boldsymbol{y}_{i:\lambda}}_{=: \boldsymbol{y}_w}$$

where

$$w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1, \quad \frac{1}{\sum_{i=1}^{\mu} w_i^2} =: \mu_w \approx \frac{\lambda}{4}$$

Auger and Hansen (2013)

Rank-one update

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, and $\mathbf{C} = \mathbf{I}$, set $\sigma = 1$, learning rate $c_{\text{cov}} \approx 2/n^2$
While not terminate

$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma \boldsymbol{y}_i, \qquad \boldsymbol{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}),$$

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \boldsymbol{y}_w \qquad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}$$

$$\mathbf{C} \leftarrow (1 - c_{\text{cov}})\mathbf{C} + c_{\text{cov}} \mu_w \underbrace{\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}}_{\text{rank-one}} \qquad \text{where } \mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \geq 1$$

Auger and Hansen (2013)

Rank-μ update

$$\begin{array}{lcllcl} \boldsymbol{x}_i & = & \boldsymbol{m} + \sigma\, \boldsymbol{y}_i, & \boldsymbol{y}_i & \sim & \mathcal{N}_i(\mathbf{0}, \mathbf{C})\,, \\ \boldsymbol{m} & \leftarrow & \boldsymbol{m} + \sigma \boldsymbol{y}_w & \boldsymbol{y}_w & = & \sum_{i=1}^{\mu} w_i\, \boldsymbol{y}_{i:\lambda} \end{array}$$

The rank-μ update extends the update rule for **large population sizes** λ using $\mu > 1$ vectors to update $\mathbf{C}$ at each generation step.
The weighted empirical covariance matrix

$$\mathbf{C}_\mu = \sum_{i=1}^{\mu} w_i\, \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^{\mathrm{T}}$$

computes a weighted mean of the outer products of the best μ steps and has rank $\min(\mu, n)$ with probability one.

with $\mu = \lambda$ weights can be negative [10]

The rank-μ update then reads

$$\mathbf{C} \leftarrow (1 - c_{\mathrm{cov}})\, \mathbf{C} + c_{\mathrm{cov}}\, \mathbf{C}_\mu$$

where $c_{\mathrm{cov}} \approx \mu_w / n^2$ and $c_{\mathrm{cov}} \leq 1$.

Auger and Hansen (2013)

Step-size update

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, evolution path $\boldsymbol{p}_\sigma = \mathbf{0}$,
set $c_\sigma \approx 4/n$, $d_\sigma \approx 1$.

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \boldsymbol{y}_w \quad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda} \qquad \text{update mean}$$

$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}}_{\text{accounts for } 1 - c_\sigma} \underbrace{\sqrt{\mu_w}}_{\text{accounts for } w_i} \boldsymbol{y}_w$$

$$\sigma \leftarrow \sigma \times \underbrace{\exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right)}_{>1 \iff \|\boldsymbol{p}_\sigma\| \text{ is greater than its expectation}} \qquad \text{update step-size}$$

Auger and Hansen (2013)

Summary of equations

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$, and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$
While not terminate

$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\,\boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \quad \text{for } i = 1, \ldots, \lambda \qquad \text{sampling}$$

$$\boldsymbol{m} \leftarrow \textstyle\sum_{i=1}^{\mu} w_i\,\boldsymbol{x}_{i:\lambda} = \boldsymbol{m} + \sigma\boldsymbol{y}_w \quad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i\,\boldsymbol{y}_{i:\lambda} \qquad \text{update mean}$$

$$\boldsymbol{p}_\mathbf{c} \leftarrow (1 - c_\mathbf{c})\,\boldsymbol{p}_\mathbf{c} + \mathbb{1}_{\{\|\boldsymbol{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_\mathbf{c})^2}\sqrt{\mu_w}\,\boldsymbol{y}_w \qquad \text{cumulation for } \mathbf{C}$$

$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\,\boldsymbol{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2}\sqrt{\mu_w}\,\mathbf{C}^{-\frac{1}{2}}\boldsymbol{y}_w \qquad \text{cumulation for } \sigma$$

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\,\mathbf{C} + c_1\,\boldsymbol{p}_\mathbf{c}\boldsymbol{p}_\mathbf{c}{}^{\mathrm{T}} + c_\mu \textstyle\sum_{i=1}^{\mu} w_i\,\boldsymbol{y}_{i:\lambda}\boldsymbol{y}_{i:\lambda}^{\mathrm{T}} \qquad \text{update } \mathbf{C}$$

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right) \qquad \text{update of } \sigma$$

Auger and Hansen (2013)

6. Syntactic Groundwater Flow Model

Synthetic Groundwater Model

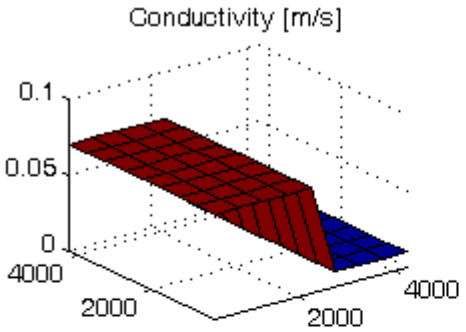
The synthetic problem is designed to amplify the search challenges and reduce the ill-posedness.

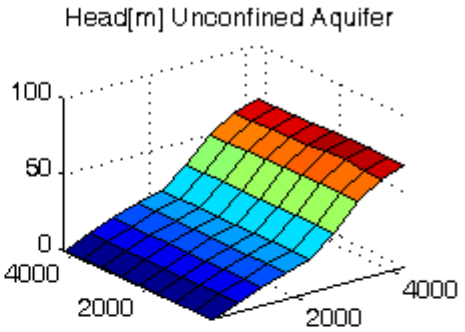
Search challenges

1. Nonlinearity: step gradient
2. Ruggedness: multimodal
3. Ill-conditioning: high curvature, so Hessian is needed.
4. Inseparability: dependency between the model parameters
5. Noise: unimportant signal to the objective function conceals the useful signal
6. High dimensionality: nonlinear increase of forward model evaluations with the increase of the number of unknowns

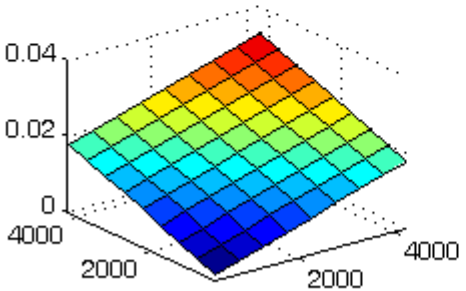
Inverse problem ill-posedness

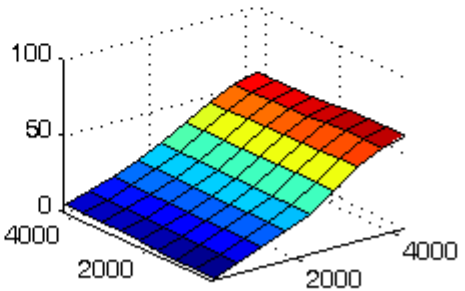
1. Nonexistence: due to observation errors
2. Nonuniqueness: due to absence of sufficient data
3. Insensitivity: low sensitivity of the state variables to model parameters
4. Instability: due to computational and observation errors





$$-\nabla.(\mathbf{K}\nabla\mathbf{h}^2) = 2\mathbf{N}$$





$$-\nabla.(\mathbf{T}\nabla\mathbf{h}) = \mathbf{N}$$

$$\min_{x\in\mathbb{R}^n} \left\| \mathbf{h}_{obs} - \mathbf{h} \right\|_2$$

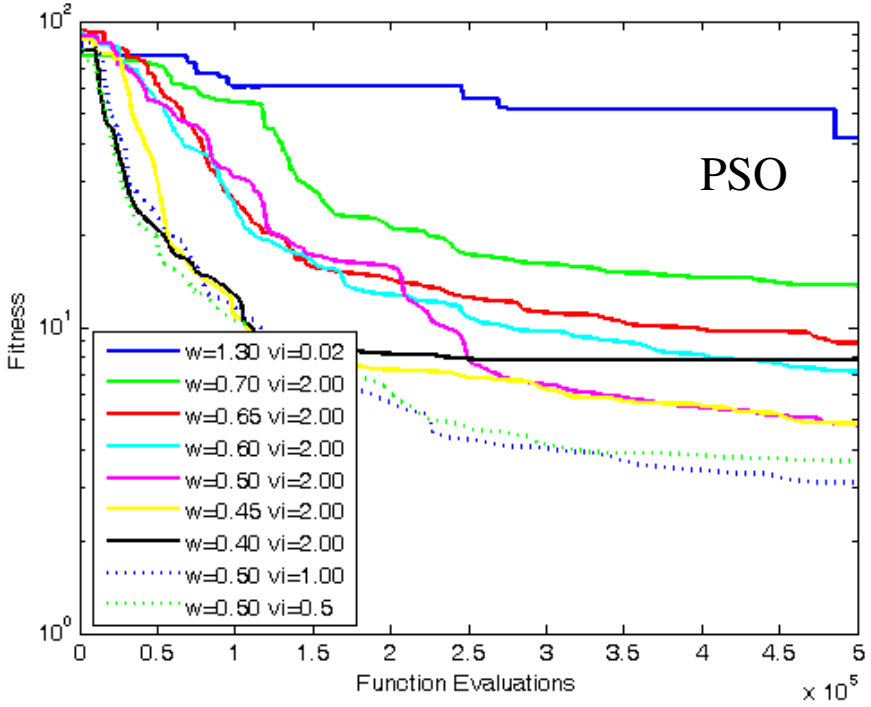
Stopping criteria
$f \le 10^{-3}$ or NFE=5×10^5

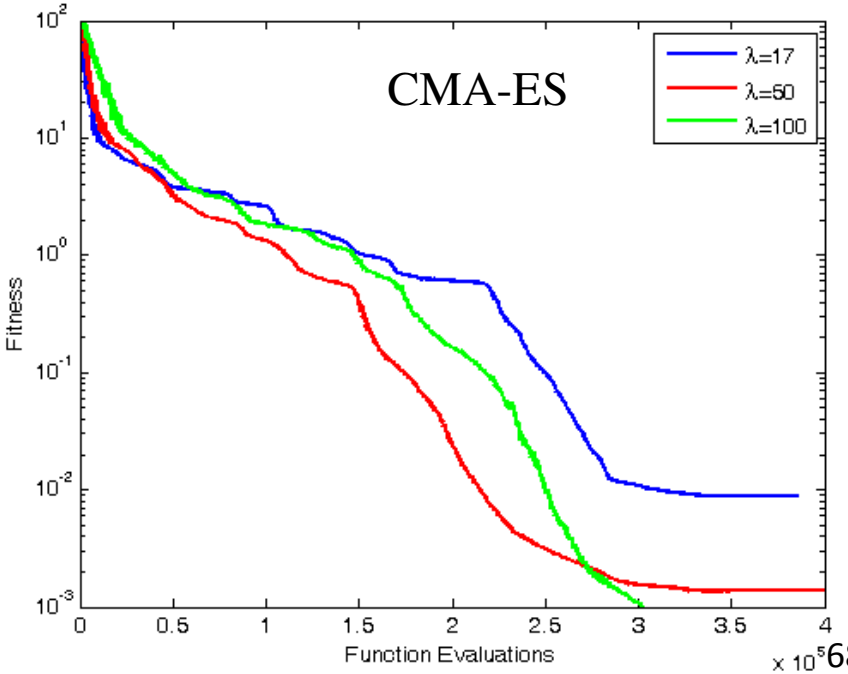
81 unknown hydraulic conductivity values
Model running time ≈ 0.1 second

Algorithm Tuning

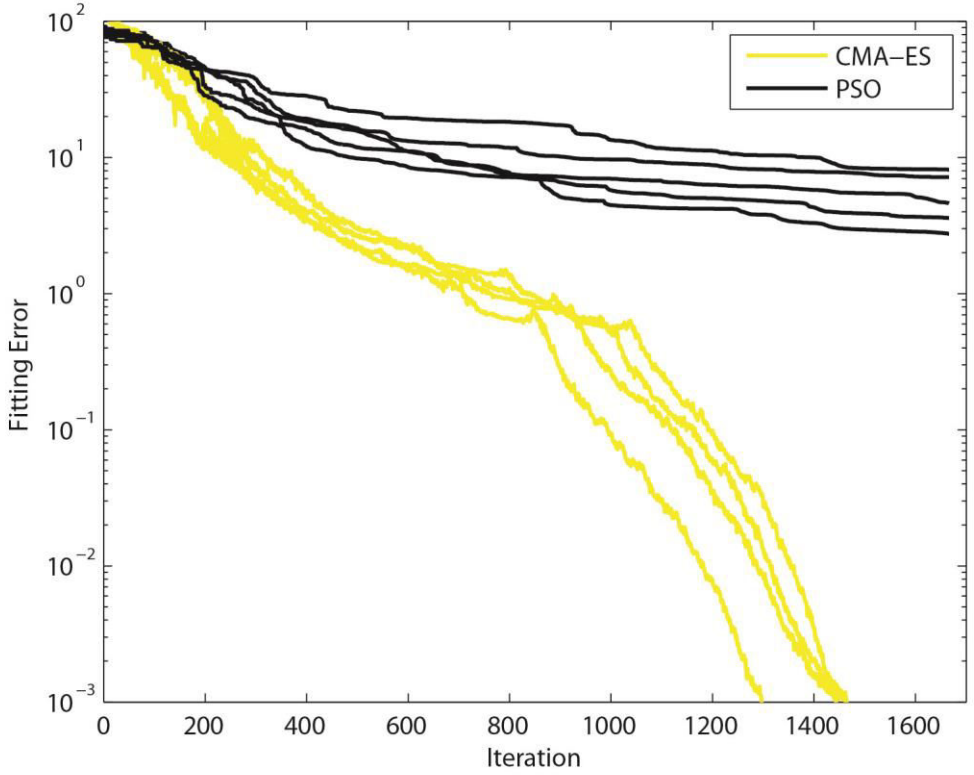
Evaluated performance of CMA-ES relative so some commonly used stochastic algorithms (Genetic Algorithm, Differential Evolution, Ant Colony and Particle Swarm Optimization) and derivative algorithm (Levenberg-Marquardt).

	Diverifcation or intenstification	Set-size control	Iteration size	Other
CMA-ES	Adaptable	Adaptable	Tunable	None
ACOR	Ranking parameter	None	Ineffective	Control parameter ζ
PSO	Particle weight Initial partical velocity	None	Tunable	No. of function evolutions
mDE	Crossover constant	None	Ineffective	None
GA	mutation rate, fraction of kept population	Adaptable	Tunable	None
L-M	Levenberg Marquardt parameter	Tunable/Adaptable	Variable	None



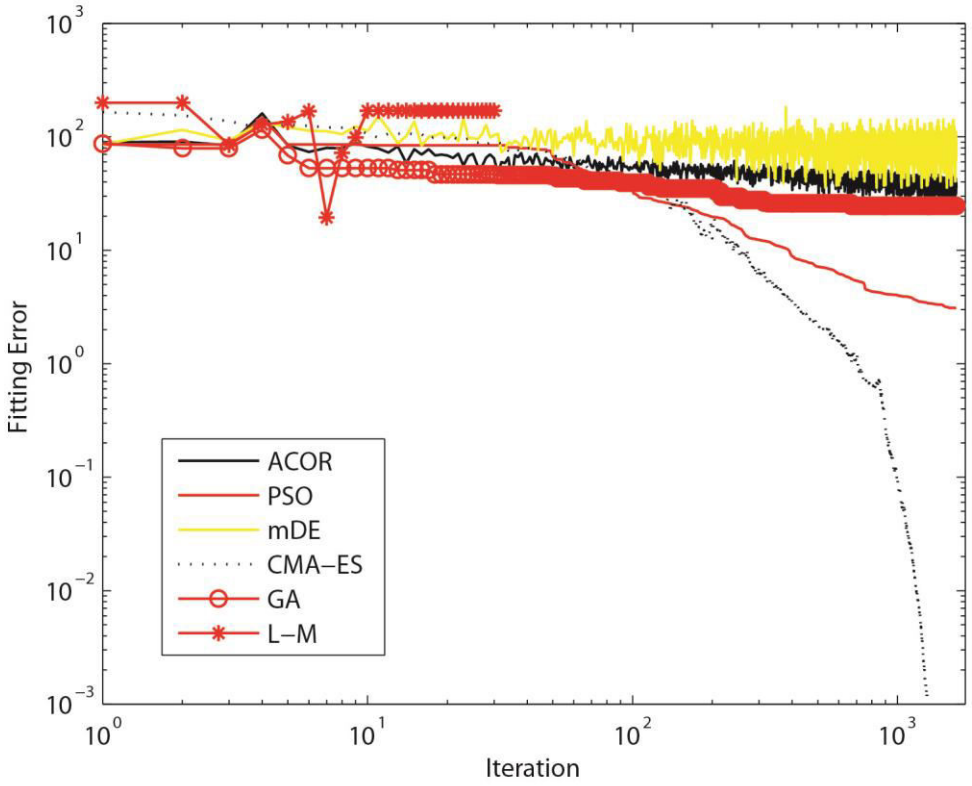


Solution Consistency



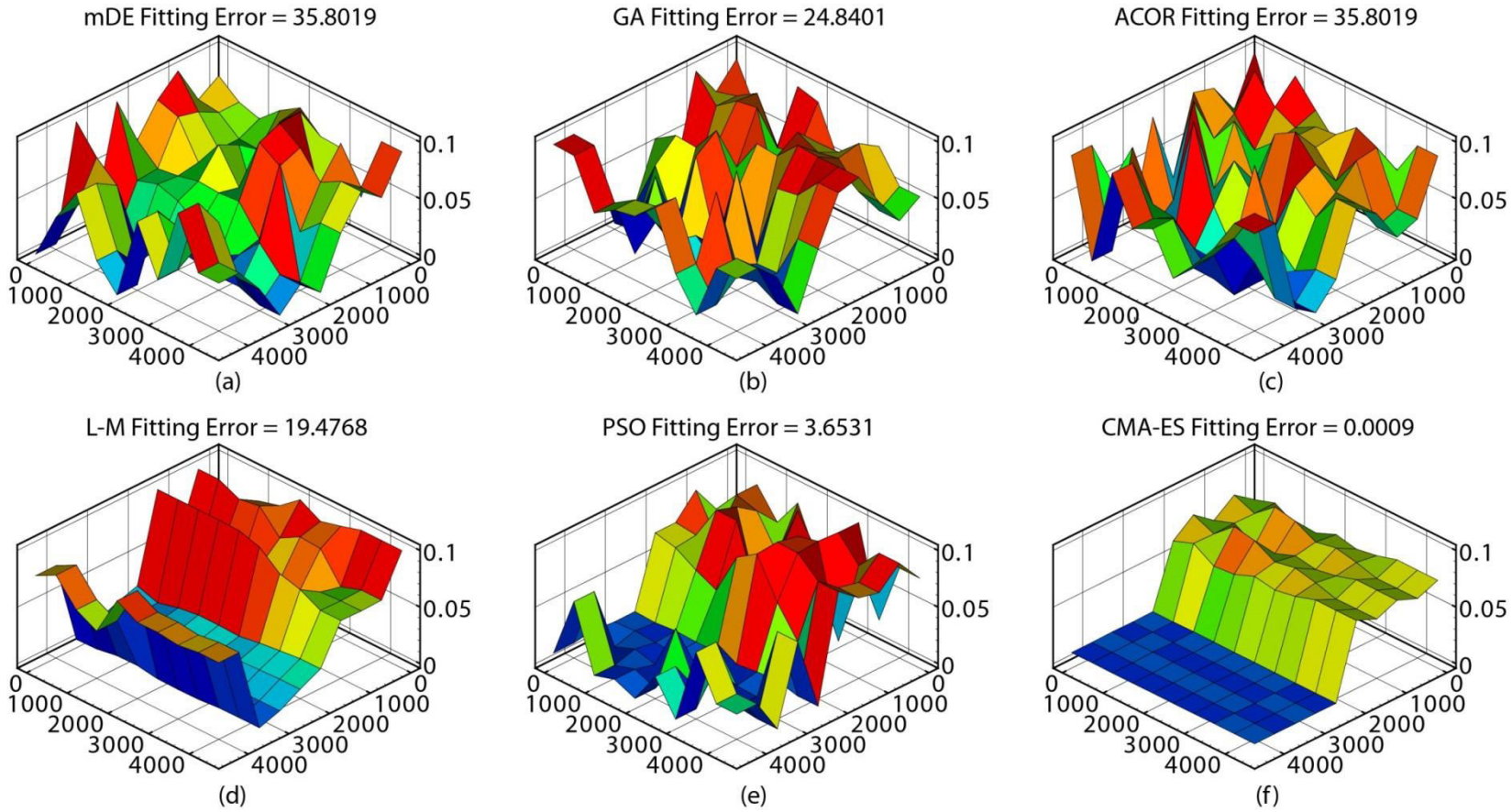
CMA-ES shows consistent results for repeated runs, which is not the case of the PSO. This is a favorable property since it eliminates the need for multiple runs to obtain the best solution.

Performance Evaluation



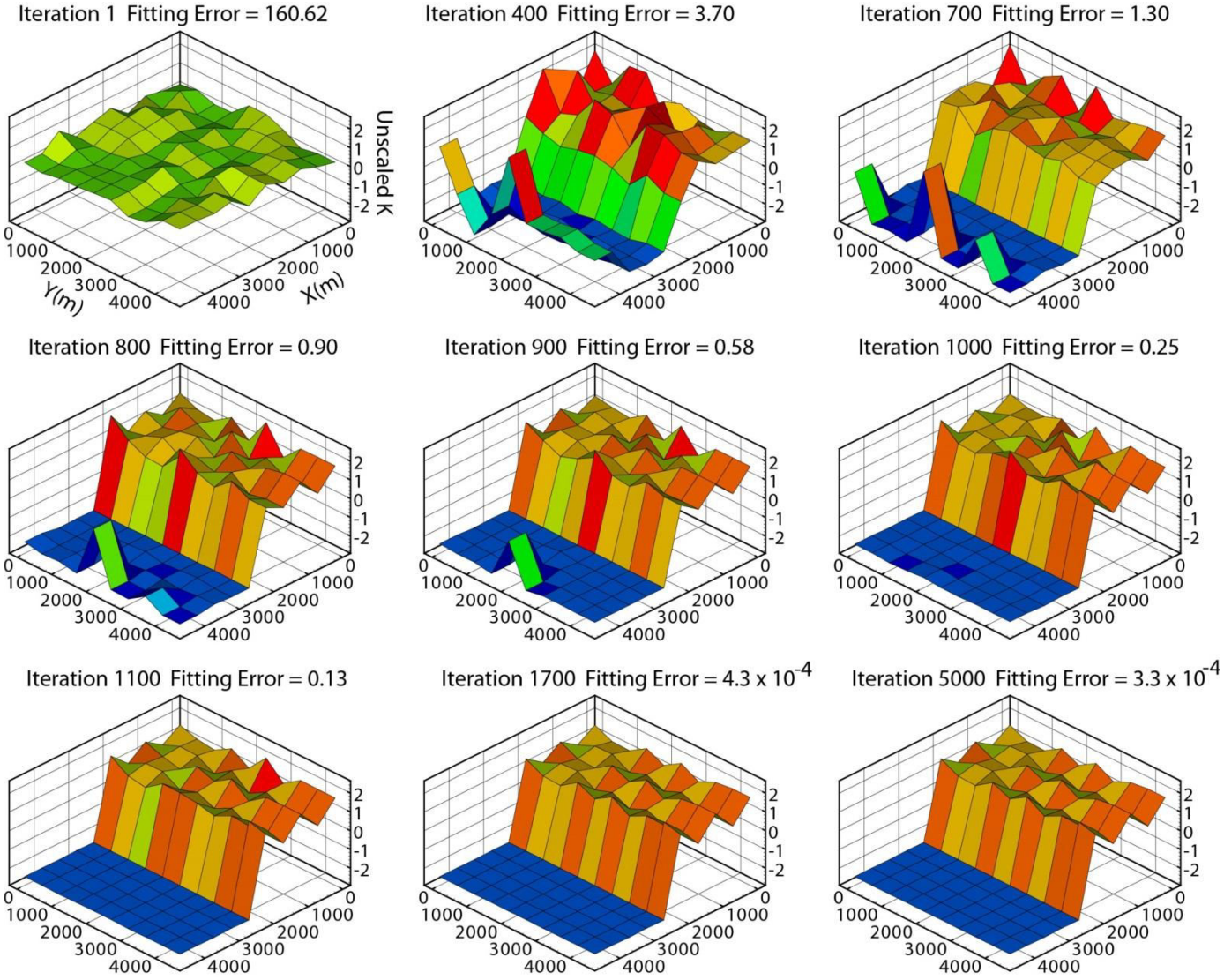
Convergence profile with iteration size of 300 except for G-N has iteration size of 90 (n+1 and the 8 additional function evaluations for the adaptation of the step size and Levenberg parameter). CMA-ES showing very good systematic local convergence, which is largely due to careful adaptation of step size.

Performance Evaluation

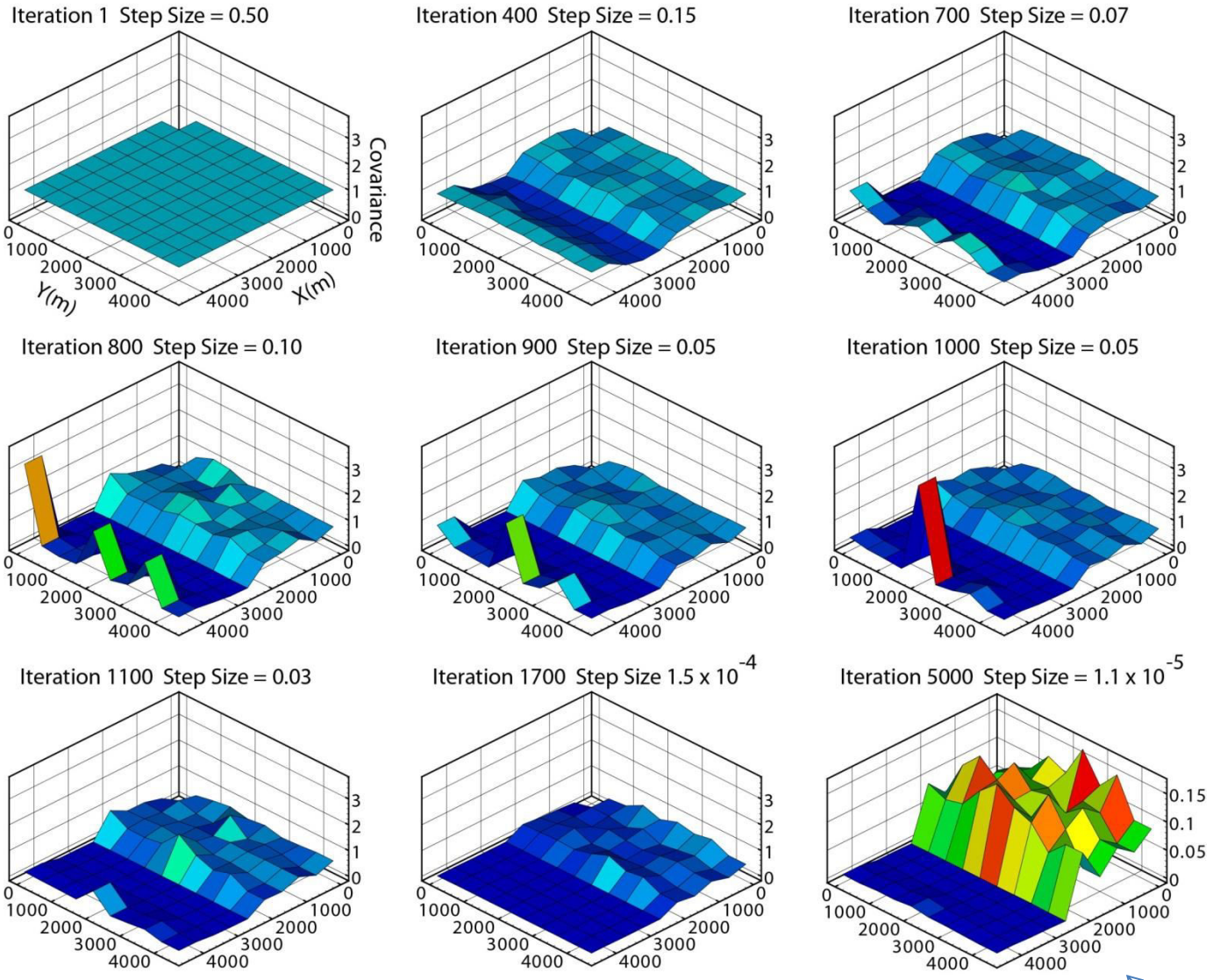


GA, ACOR and mDE failed to recognize the structure of the hydraulic conductivity field. The second and third best performance is of PSO and L-M, respectively. Yet PSO and L-M were deceived by the noise. Yet unlike G-N, PSO cannot effectively exploit correlation between variables, so the zone of higher conductivity did not smooth-out. CMA-ES overcame these two pitfalls due to the utilization of a second order of learning through the adaptation of the covariance matrix, reaching a near global solution unlike all other algorithms.

Performance Evaluation



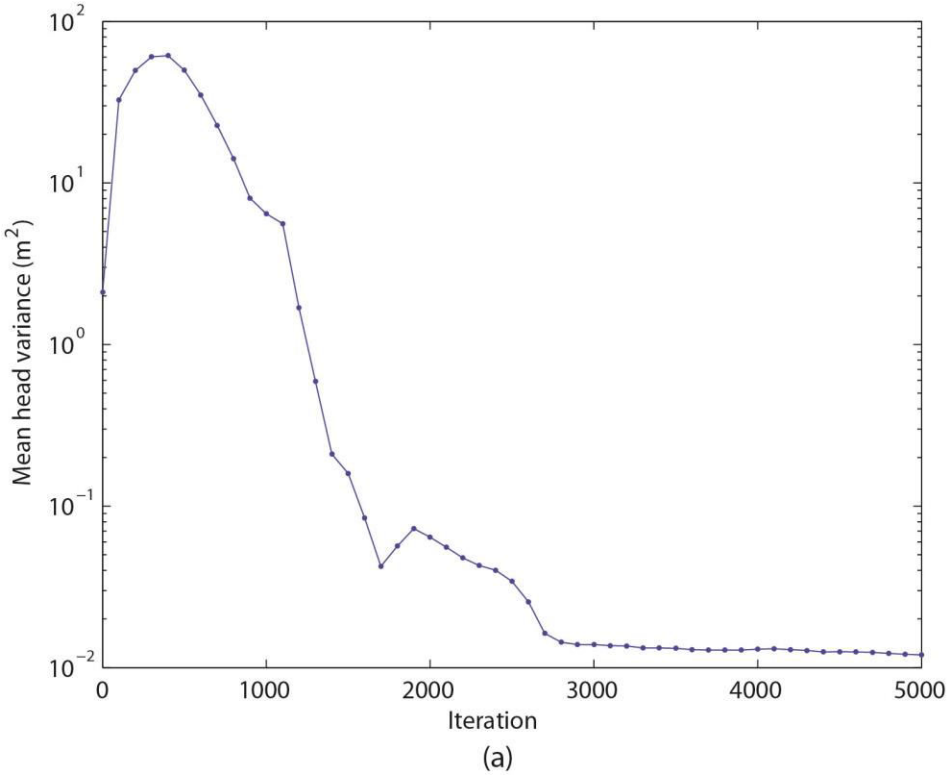
Performance Evaluation

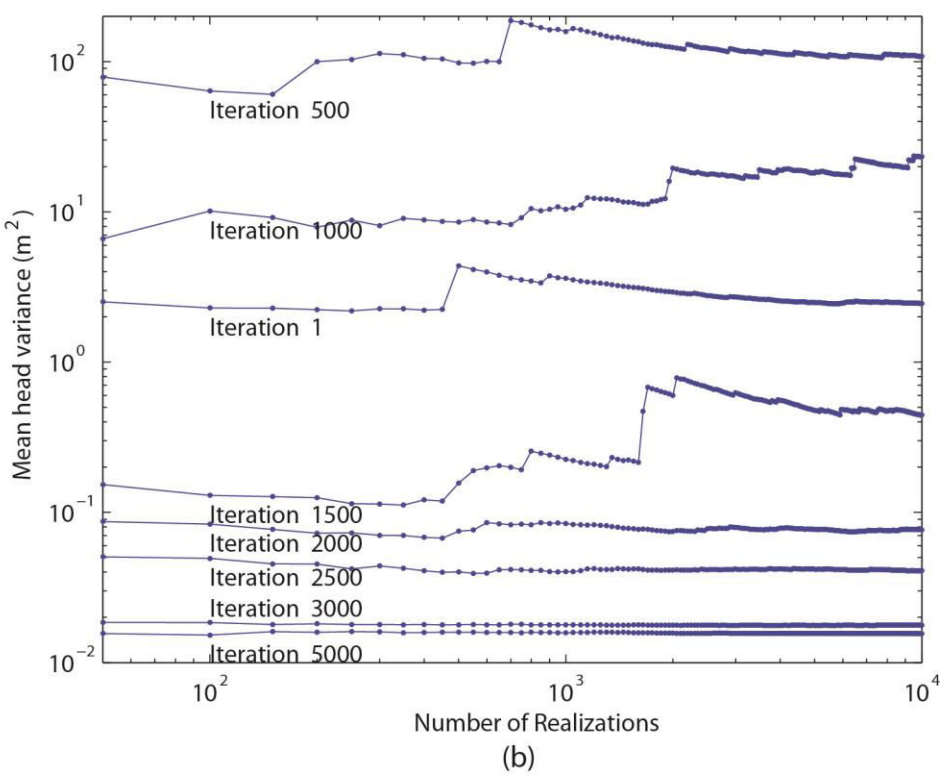


Diagonal elements: variances

Has a different scale

Covariance Matrix Monte Carlo Simulation





(a) Mean head variance for the unconfined aquifer based on 10,000 realizations at different sampling intervals shows that the variance is not necessarily monotonically decreasing.

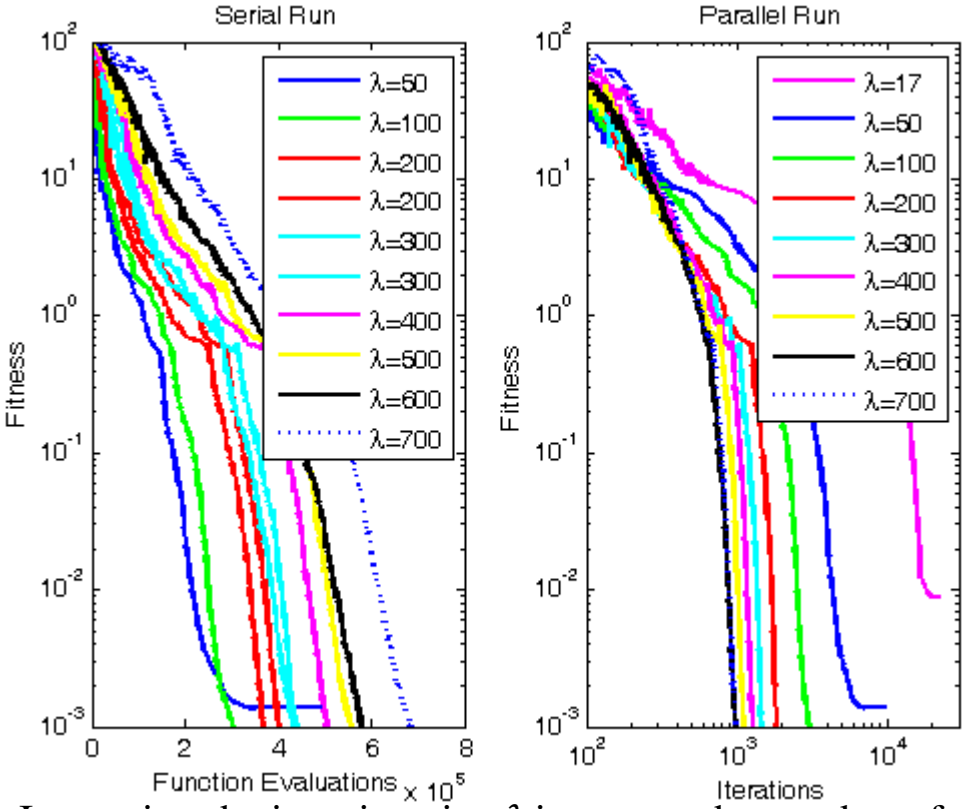
(b) Convergence profiles of mean head variance for the unconfined aquifer for different sampling intervals shows after convergence, very few number of samples are required.

Parallel Performance

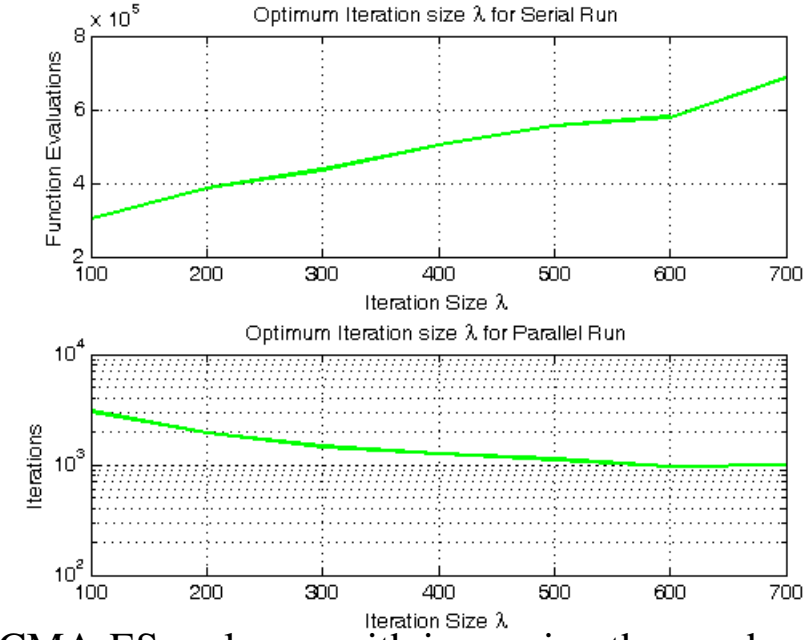
CMA-ES is readily amendable for embracing parallel master-slave technique with minimal overhead since there is no communication between solutions (i.e. models with different parameter sets)

Iteration size λ is the **number of solutions** per iteration

Number of solutions is equal to **the number of cores**



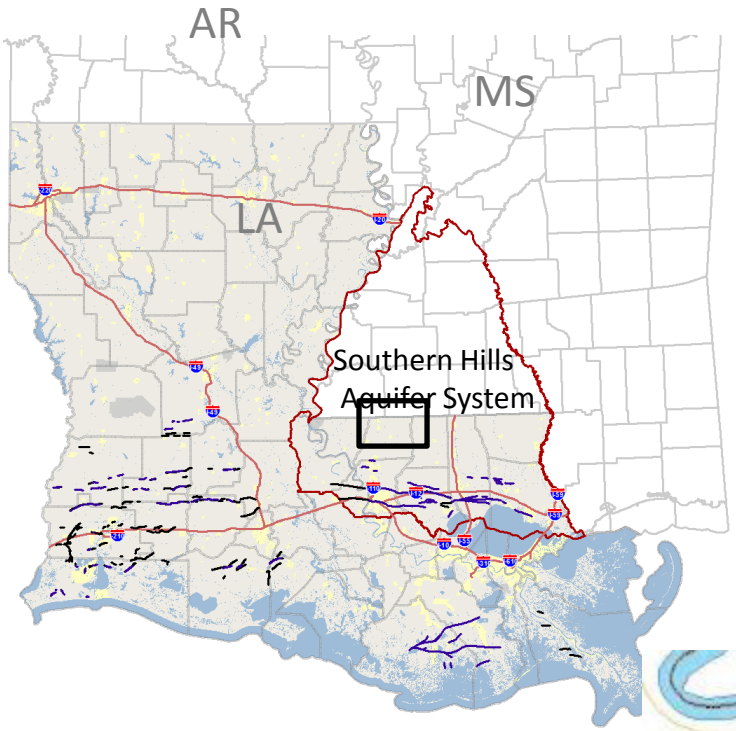
Increasing the iteration size λ increases the number of forward model evaluations, while significantly decreasing the number of iterations. This means that CMA-ES scales up with increasing the number of cores.

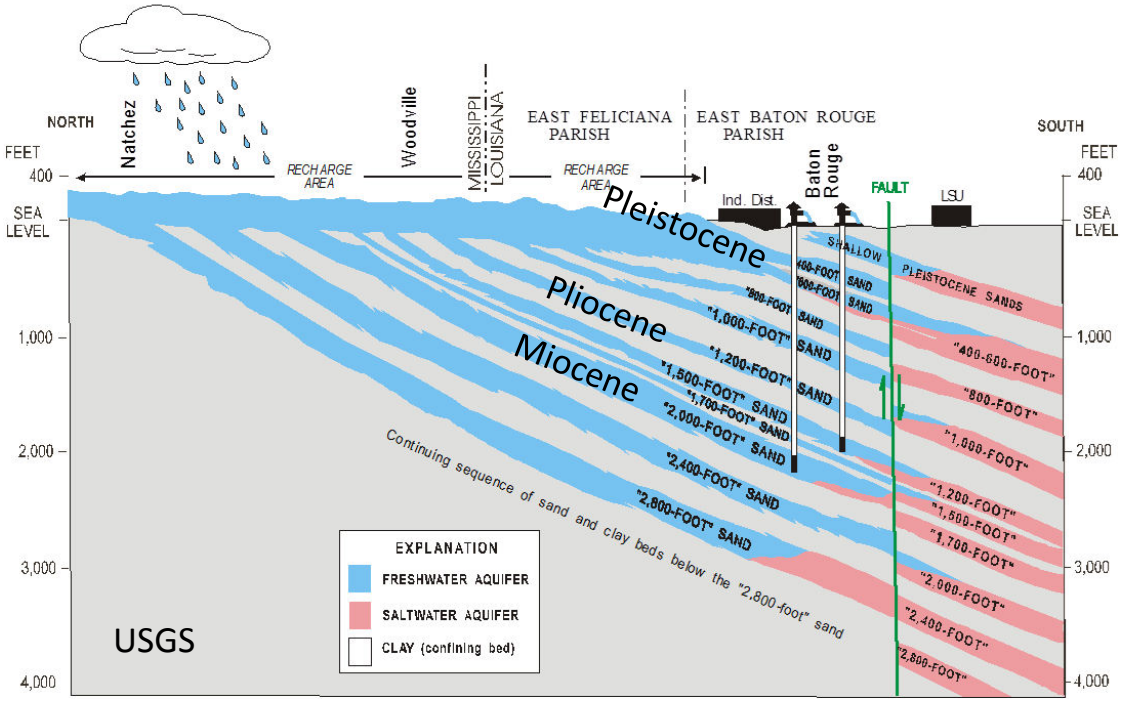


CMA-ES scales up with increasing the number of cores up to a certain point. Number of iterations for λ=100 is 2109, for λ=600 is 964, and for λ=700 is 984. This observation is consistent with Hansen and Kern [2004] results on eight test functions, which show that the scaling has a convex shape.

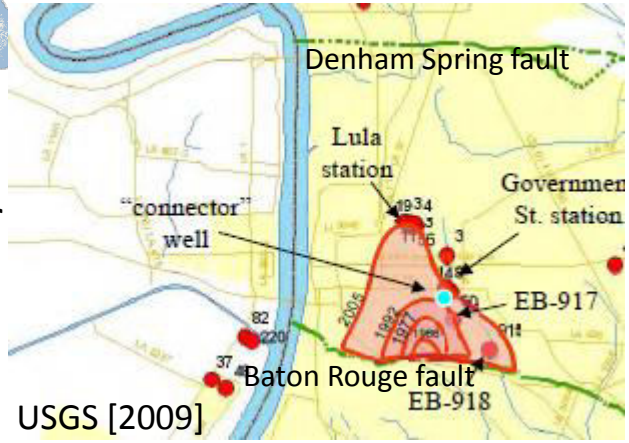
7. Baton Rouge Groundwater Flow Models

Baton Rouge salt water intrusion problem

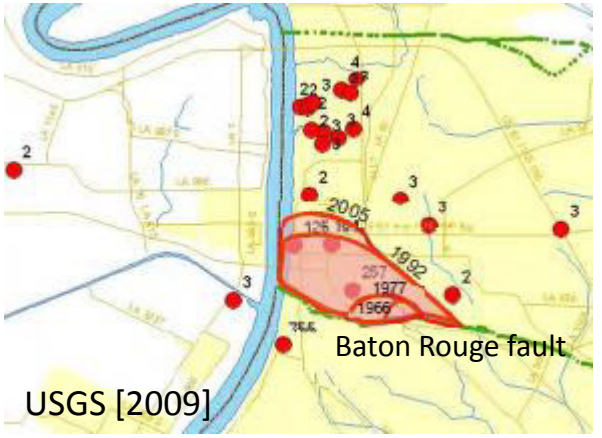




Motivation: Municipal and Industrial well supply are under salinization threat [USGS,2009]



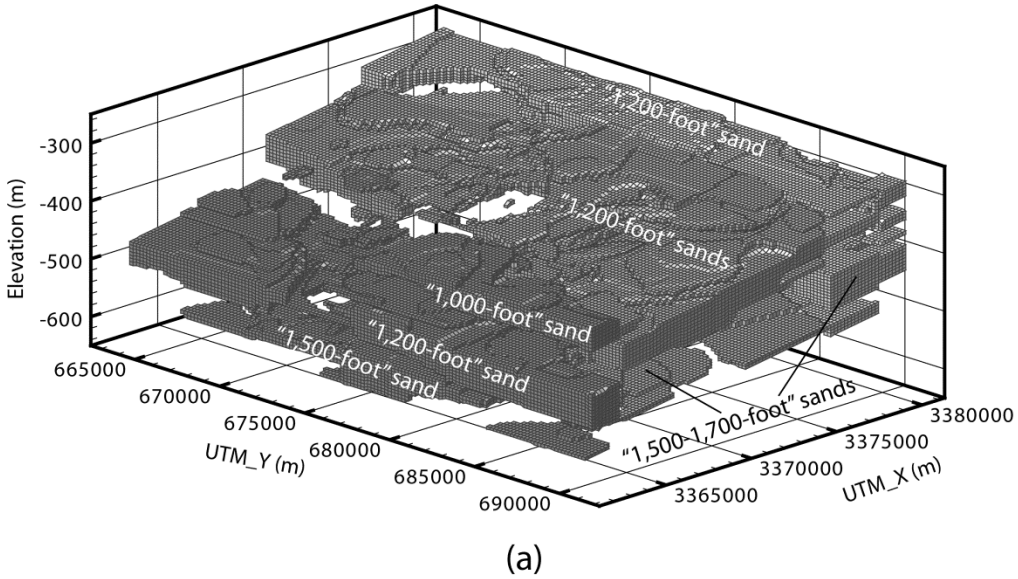
Brackish water plume "1500-foot" Sand



Brackish water plume "2000-foot" Sand

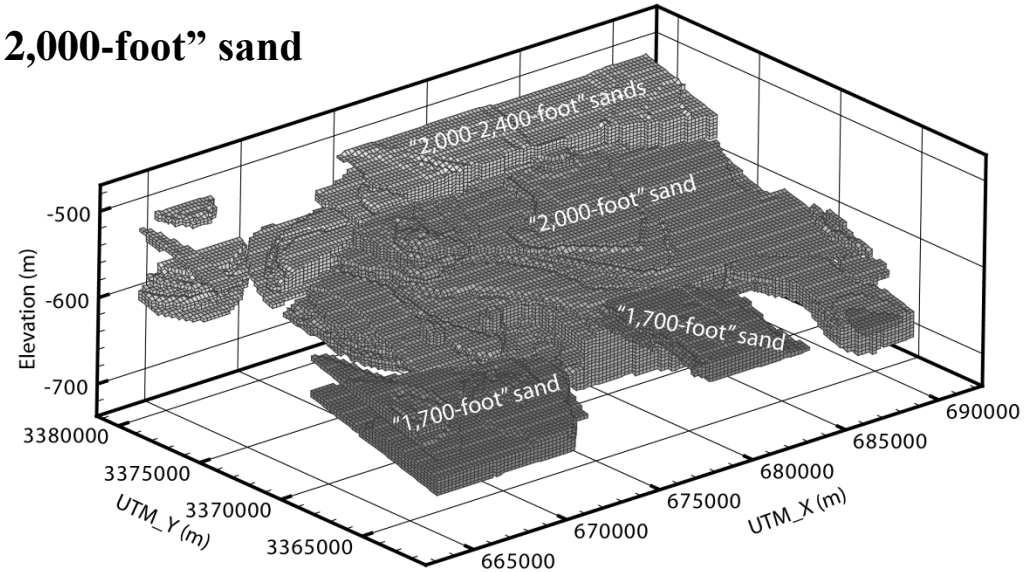
Baton Rouge Groundwater Flow Models

“1,200-1,500-1,700-foot” sands



(a)

2,000-foot” sand

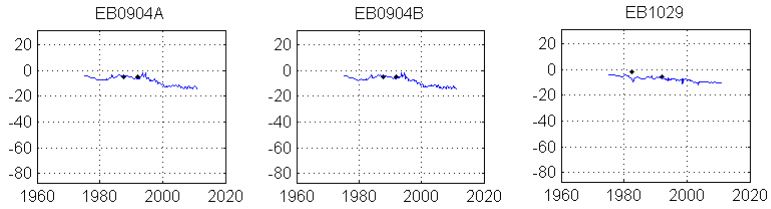


(b)

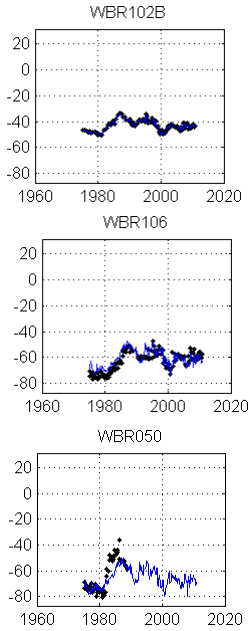
Flow model calibration:

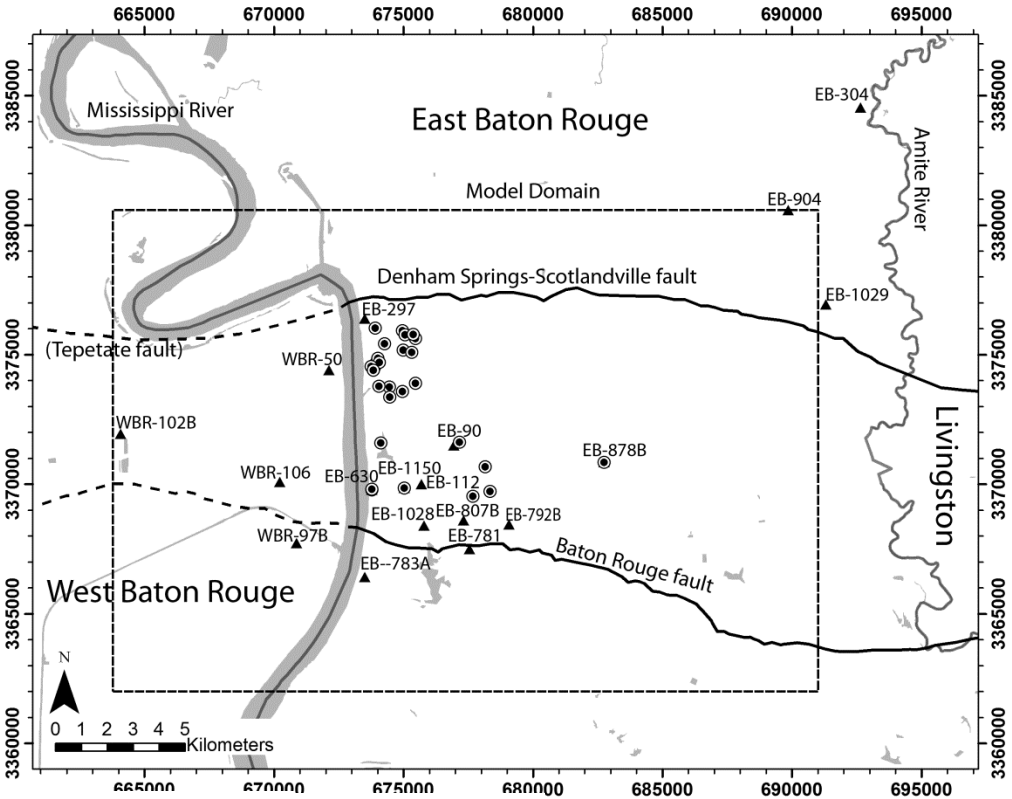
- 35-year head data from the USGS
- Detailed pumpage data from Louisiana Capital Area Ground Water Conservation Commission
- Detailed sedimentary architecture that accounts for the Baton Rouge fault system
- Each layer has 12,741 cells and each cell is 200x200m
- Layers have variable thickness ~ 4 to 6 m.
- The “1,200-1,500-1,700-foot” sands has 45 layers. The “2,000-foot” sand has 29 layers

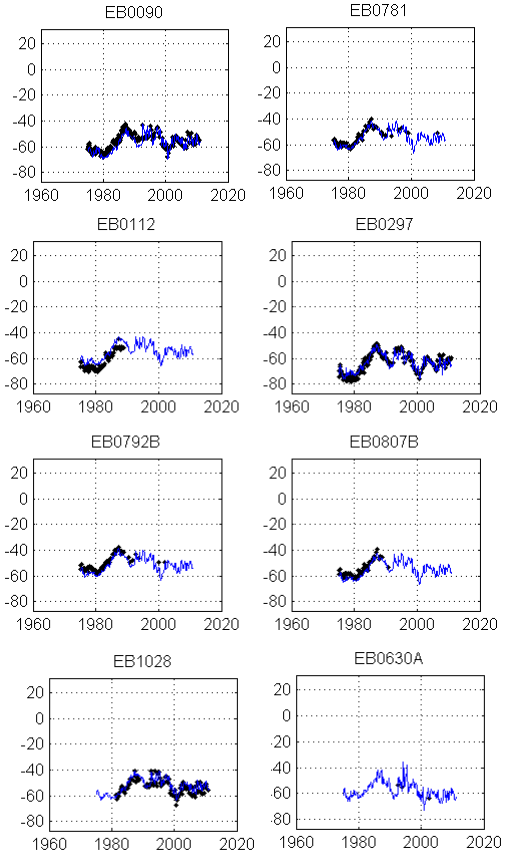
Groundwater model calibration

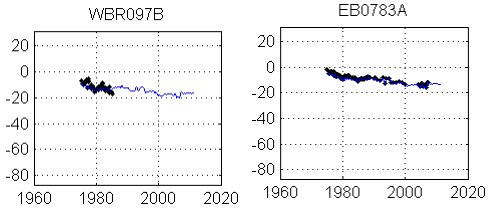


Easy calibration with 6 unknown parameters





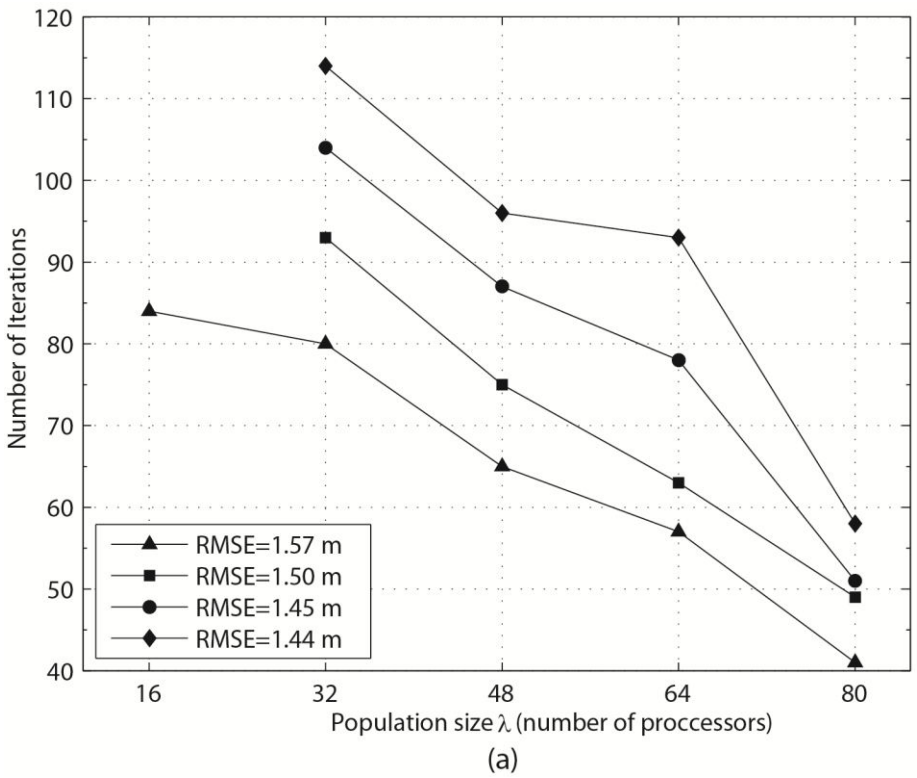


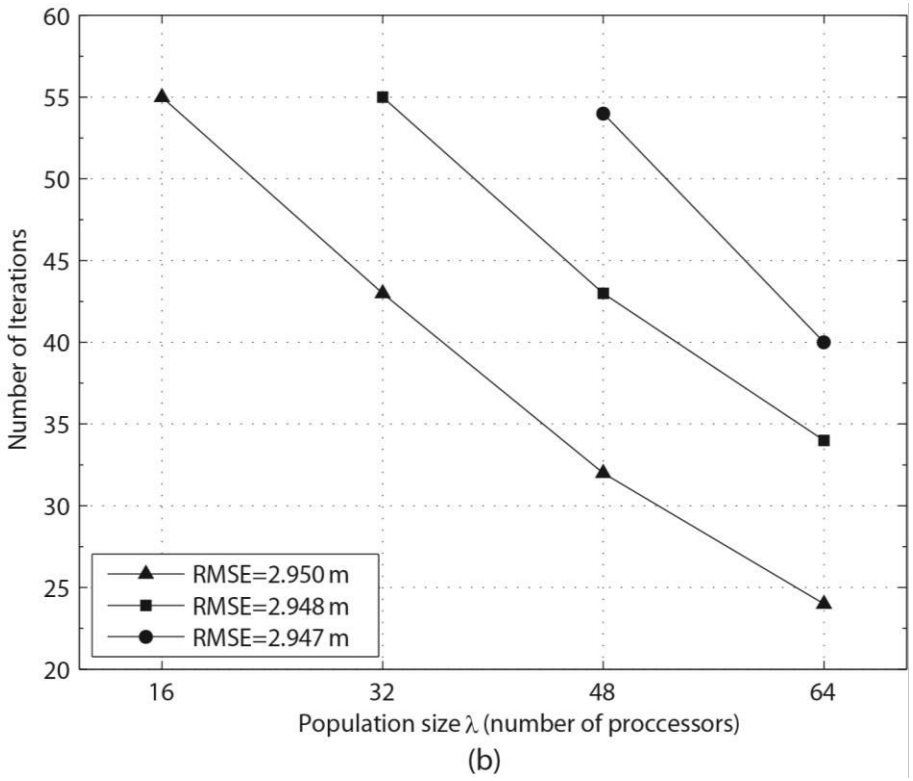


Fitting error =2.94 m

Acknowledgement Model calibration was carried out using SuperMike-II cluster at LSU HPC

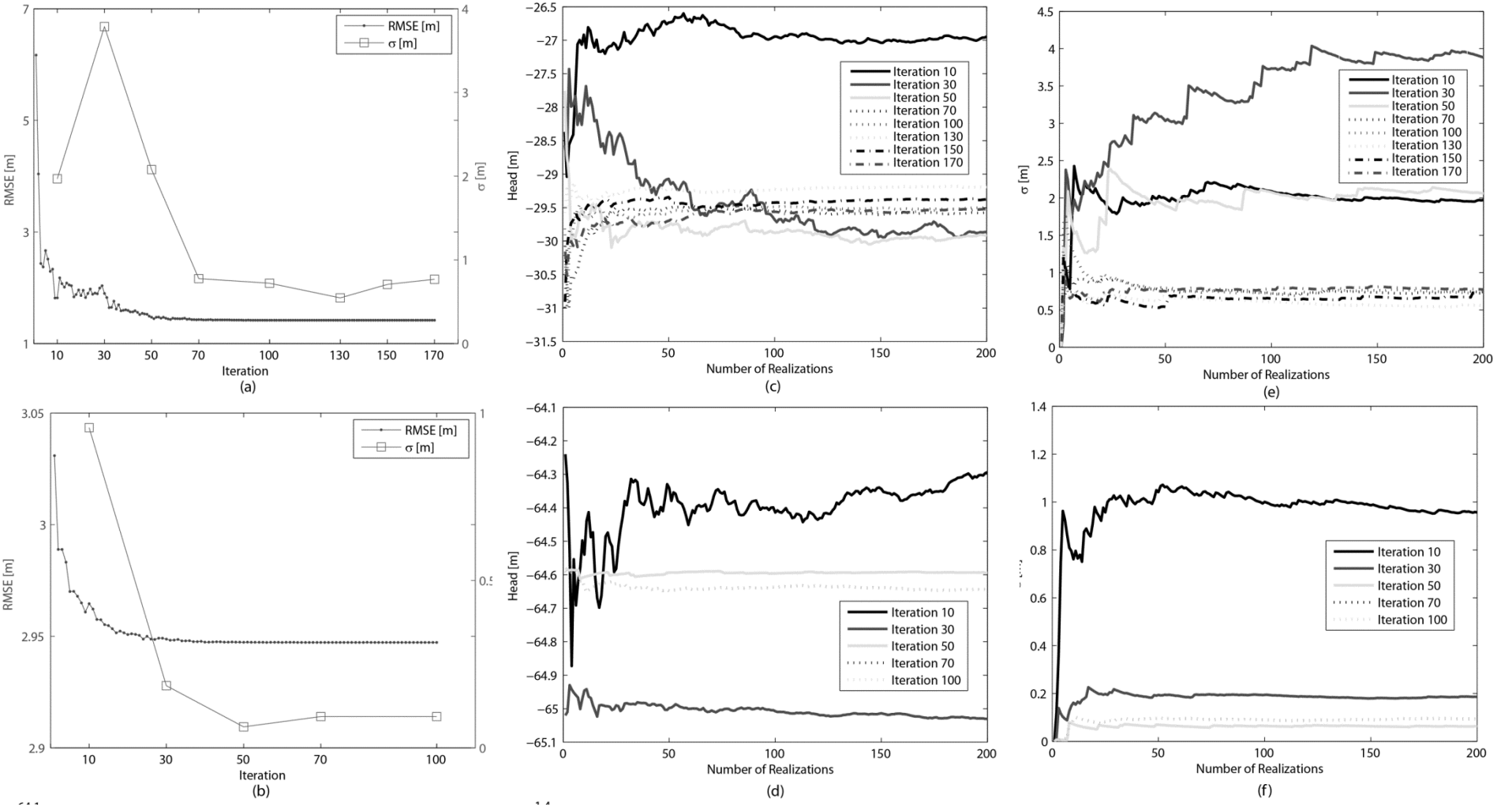
Evaluating Parallel Performance





The number of iterations for different population sizes required to reach several target fitting errors for the (a) "1,200-1,500-1,700-foot" sands model and (b) "2,000-foot" sand model.

Covariance Matrix Monte Carlo Simulation



The convergence profiles of the RMSE [m] and head standard deviation [m] for a selected observation point based on 200 realizations at successive sampling intervals for (a) the "1,200-1,500-1,700-foot" sands model and (b) the "2,000-foot" sands model. Head [m] convergence profiles at successive sampling intervals for (c) the "1,200-1,500-1,700-foot" sands model and (d) the "2,000-foot" sands model. Head standard deviation [m] convergence profiles at successive sampling intervals for (e) the "1,200-1,500-1,700-foot" sands model and (f) the "2,000-foot" sands model. Note that for subplots (d) and (f) the convergence profiles for iteration 70 and 100 are similar.

Conclusions

- CMA-ES is a powerful and very practical tool for inverse groundwater modeling and Monte Carlo simulation (only one parameter to tune, reach a near global solution, exhibits solution consistency for repeated runs, shows favorable scaling with increasing the number of processors for parallel run, invariance properties, very efficient multivariate Monte Carlo sampling).

- Parallel CMA-ES encourages the development of realistic groundwater model due to the significant alleviation of the computational cost.

- Empirically estimated covariance matrix is precise and can be used for Monte Carlo sampling to quantify parameter related uncertainty with relatively very few number of samples.

- The quantified variance is only due to parameter estimation error, which it is a measure of the precision of the solution (regardless of course of the adequacy of the model).

References

Many of the materials in this presentation are obtained from the following sources

[1] Anne Auger and Nikolaus Hansen (2013), Tutorial CMA-ES (Covariance Matrix Adaptation Evolution Strategies), GECCO 2013, July, 2013, Amsterdam,

[2] Marc Toussaint (2013), Introduction to Optimization Course, Summer Term 2013, University of Stuttgart

[3] Nikolaus Hansen (2011), CMA-ES – a Stochastic Second-Order Method for Function-Value Free Numerical Optimization, Microsoft Research Cambridge, Lecture, November 2011

[4] Nikolaus Hansen (2010), CMA-Evolution Strategy: A Tutorial.

[5] Hansen, N., D.V. Arnold, and A. Auger (2013). Evolution Strategies. To appear in Janusz Kacprzyk and Witold Pedrycz (Eds.): Handbook of Computational Intelligence, Springer

Thank You