Introduction to Optimization

Lecture 4: Continuous Optimization III

(stochastic algorithms, derivative-free optimization, critical performance assessment)

October 18, 2019

TC2 - Optimisation

Université Paris-Saclay

Anne Auger and Dimo Brockhoff

Inria Saclay – Ile-de-France

Course Overview

Date		Topic
Fri, 27.9.2019	DB	Introduction
Fri, 4.10.2019 (4hrs)	AA	Continuous Optimization I: differentiability, gradients, convexity, optimality conditions
Fri, 11.10.2019 (4hrs)	AA	Continuous Optimization II: constrained optimization, gradient-based algorithms, stochastic gradient
Fri, 18.10.2019 (4hrs)	DB	Continuous Optimization III: stochastic algorithms, derivative-free optimization, critical performance assessment [1st written test]
Wed, 30.10.2019	DB	Discrete Optimization I: graph theory, greedy algorithms
Fri, 15.11.2019	DB	Discrete Optimization II: dynamic programming, heuristics [2nd written test]
Fri, 22.11.2018		final exam

Details on Continuous Optimization Lectures

Introduction to Continuous Optimization

- examples (from ML / black-box problems)
- typical difficulties in optimization

Mathematical Tools to Characterize Optima

- reminders about differentiability, gradient, Hessian matrix
- unconstraint optimization
    - first and second order conditions
    - convexity

- constraint optimization

Gradient-based Algorithms

- quasi-Newton method (BFGS)

DFO: trust-region method (Nelder-Mead)

Learning in Optimization / Stochastic Optimization

- CMA-ES (adaptive algorithms / Information Geometry)
- PhD thesis possible on this topic

method strongly related to ML / new promising research area
interesting open questions

CMA-ES in a Nutshell

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_c = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
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

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

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

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

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

$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\,\mathbf{C} + c_1 \boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}} + \ldots$

$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathrm{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\ldots\right.$

Not covered on this slide: terminati...
encoding

Goal:
Understand the main principles of this state-of-the-art algorithm.

Copyright Notice

- Last slide was taken from **`http://www.cmap.polytechnique.fr/~nikolaus.hansen/copenhagen-cma-es.pdf`** (copyright by Nikolaus Hansen, one of the main inventors of the CMA-ES algorithms)
- In the following, I will borrow more slides from there and from **`http://www.cmap.polytechnique.fr/~dimo.brockhoff/optimizationSaclay/2015/slides/20151106-continuousoptIV.pdf`**
(by Anne Auger)

- In the following and the online material in particular, I refer to these pdfs as [Hansen, p. X] and [Auger, p. Y] respectively.

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_c = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
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

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

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

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

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

$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\,\mathbf{C} + c_1 \boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}} + \ldots$ — update C

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

Not covered on this slide: terminati…
encoding

Goal:
Understand the main principles of this state-of-the-art algorithm.

CMA-ES: Stochastic Search Template

A stochastic blackbox search template to minimize $f: \mathbb{R}^n \to \mathbb{R}$

Initialize distribution parameters θ, set population size $\lambda \in \mathbb{N}$

While happy do:

- Sample distribution $P(\boldsymbol{x}|\theta) \to \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda \in \mathbb{R}^n$
- Evaluate $\boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda$ on f
- Update parameters $\theta \leftarrow F_\theta(\theta, \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda, f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_\lambda))$

For CMA-ES and evolution strategies in general:

sample distributions = multivariate Gaussian distributions

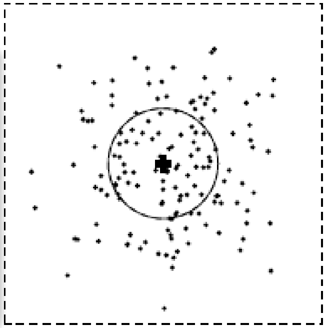
Sampling New Candidate Solutions (Offspring)

Evolution Strategies

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

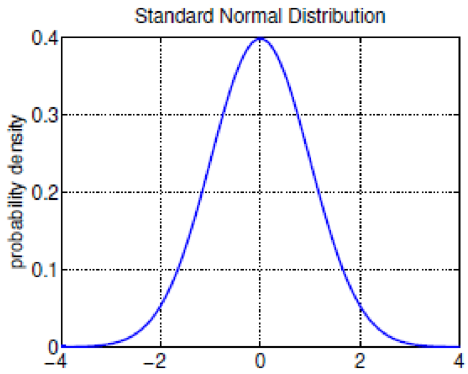
here, all new points are sampled with the same parameters

it remains to show how to adapt the parameters, but for now: normal distributions

from [Auger, p. 10]

Normal Distribution

1-D case



probability density of the 1-D standard normal distribution $\mathcal{N}(0,1)$

(expected (mean) value, variance) = (0,1)

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

General case

- Normal distribution $\mathcal{N}(\boldsymbol{m}, \sigma^2)$

  (expected value, variance) = $(\boldsymbol{m}, \sigma^2)$

  density: $p_{\boldsymbol{m},\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\boldsymbol{m})^2}{2\sigma^2}\right)$
- A normal distribution is entirely determined by its mean value and variance
- The family of normal distributions is closed under linear transformations: if X is normally distributed then a linear transformation $aX + b$ is also normally distributed
- **Exercice:** Show that $\boldsymbol{m} + \sigma\mathcal{N}(0,1) = \mathcal{N}(\boldsymbol{m}, \sigma^2)$

from [Auger, p. 11]

Normal Distribution

General case

A random variable following a 1-D normal distribution is determined by its mean value $\boldsymbol{m}$ and variance σ^2.

In the n-dimensional case it is determined by its mean vector and covariance matrix

Covariance Matrix

If the entries in a vector $\boldsymbol{X} = (X_1, \dots, X_n)^T$ are random variables, each with finite variance, then the covariance matrix Σ is the matrix whose (i, j) entries are the covariance of (X_i, X_j)

$$\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$

where $\mu_i = \mathrm{E}(X_i)$. Considering the expectation of a matrix as the expectation of each entry, we have

$$\Sigma = \mathrm{E}[(X - \mu)(X - \mu)^T]$$

Σ is symmetric, positive definite

from [Auger, p. 12]

The Multi-Variate (n-Dimensional) Normal Distribution

Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

density: $p_{\mathcal{N}(\boldsymbol{m},\mathbf{C})}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right),$

from [Auger, p. 13]

The Multi-Variate (n-Dimensional) Normal Distribution

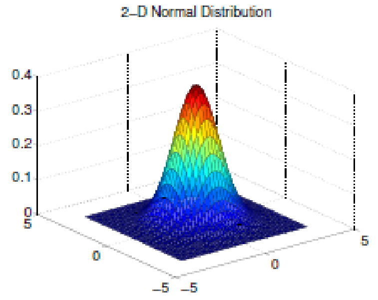
Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

$$\text{density: } p_{\mathcal{N}(\boldsymbol{m},\mathbf{C})}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x} - \boldsymbol{m})\right),$$

The mean value $\boldsymbol{m}$

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

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



from [Auger, p. 13]

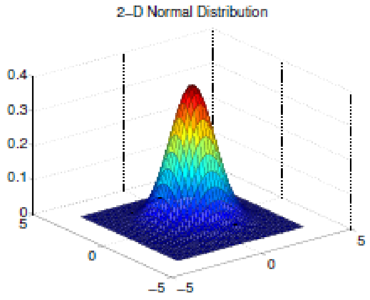
The Multi-Variate (n-Dimensional) Normal Distribution

Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

$$\text{density: } p_{\mathcal{N}(\boldsymbol{m},\mathbf{C})}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right),$$

The mean value $\boldsymbol{m}$

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



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

The covariance matrix $\mathbf{C}$

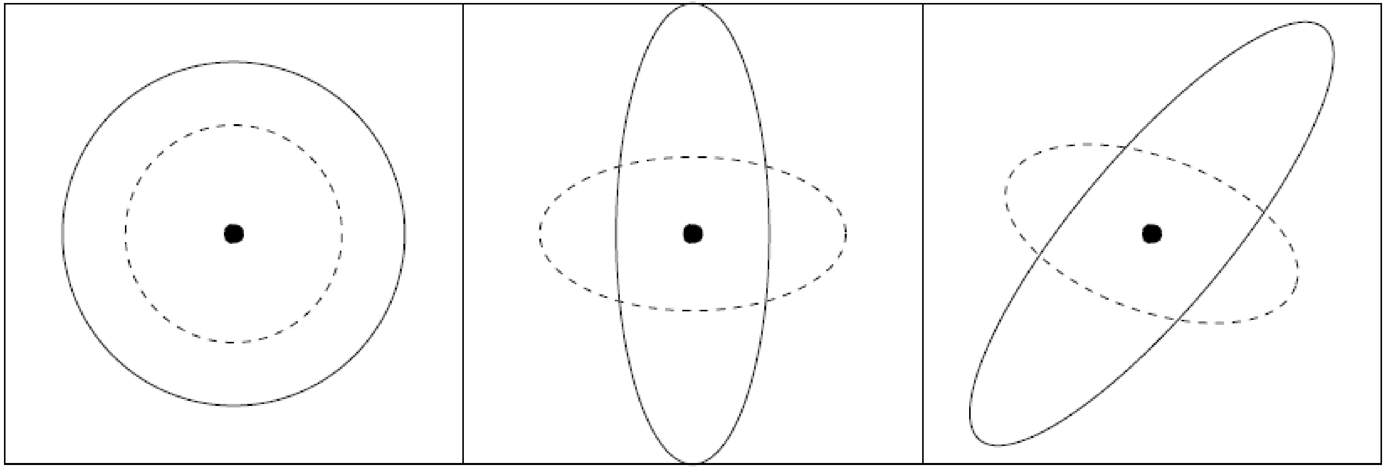
- determines the shape
- geometrical interpretation: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, (\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m}) = 1\}$

from [Auger, p. 13]

Covariance Matrix: Lines of Equal Density

...any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \mid (\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}} \mathbf{C}^{-1} (\boldsymbol{x} - \boldsymbol{m}) = 1\}$

Lines of Equal Density



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

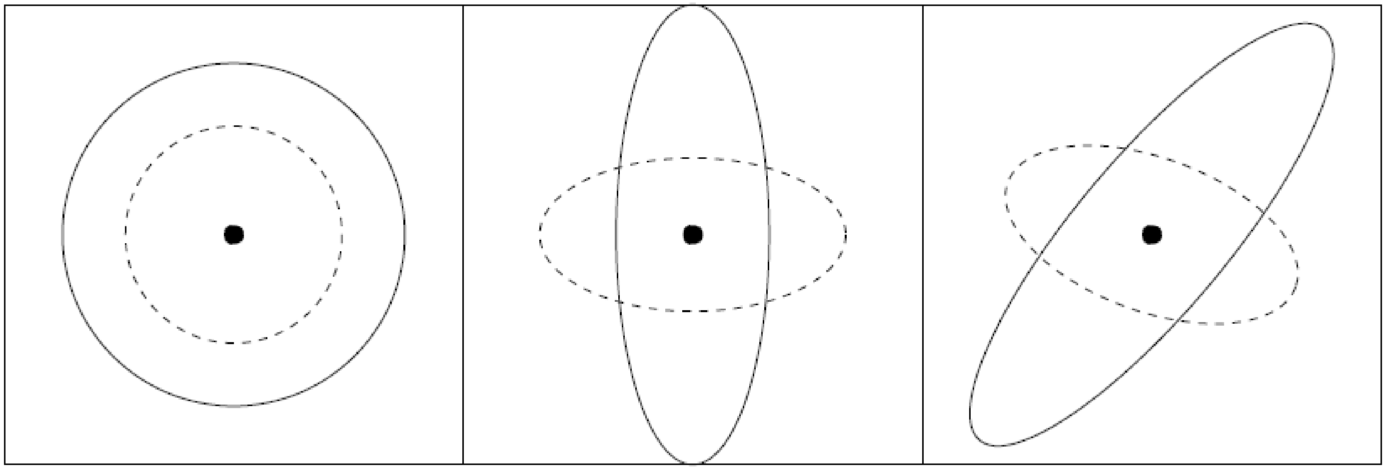
one degree of freedom σ

components are independent standard normally distributed

where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

…any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{x \in \mathbb{R}^n \mid (x - m)^{\mathrm{T}} \mathbf{C}^{-1}(x - m) = 1\}$



$\mathcal{N}(m, \sigma^2\mathbf{I}) \sim m + \sigma\mathcal{N}(\mathbf{0}, \mathbf{I})$
one degree of freedom σ
components are independent standard normally distributed

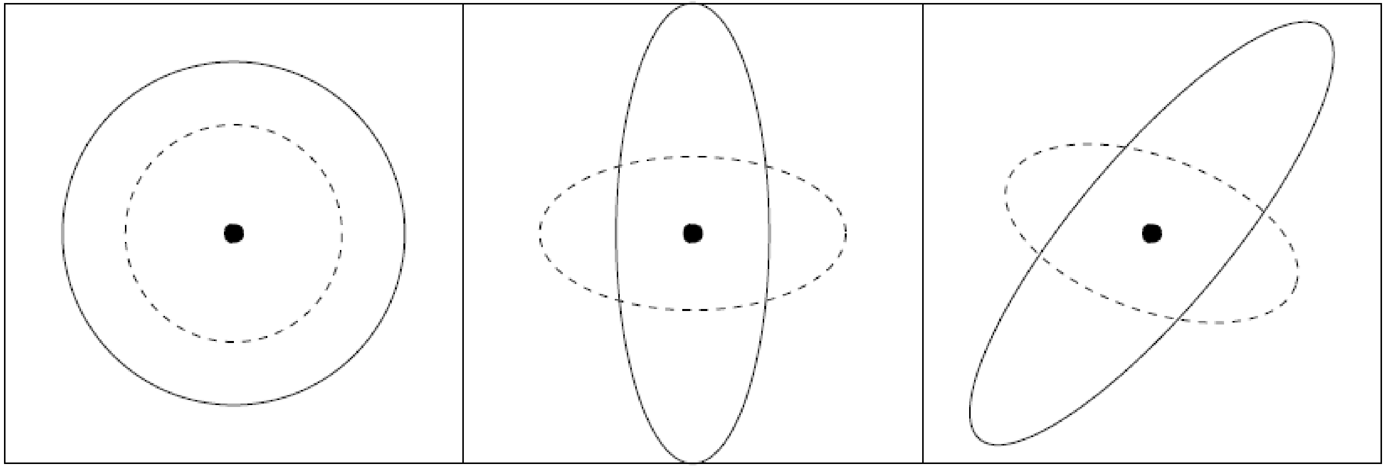
$\mathcal{N}(m, \mathbf{D}^2) \sim m + \mathbf{D}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$
n degrees of freedom
components are independent, scaled

where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

...any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{x \in \mathbb{R}^n \mid (x - m)^{\mathrm{T}} \mathbf{C}^{-1} (x - m) = 1\}$

Lines of Equal Density



$\mathcal{N}(m, \sigma^2 \mathbf{I}) \sim m + \sigma \mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(m, \mathbf{D}^2) \sim m + \mathbf{D} \mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(m, \mathbf{C}) \sim m + \mathbf{C}^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I})$
one degree of freedom σ	n degrees of freedom	$(n^2 + n)/2$ degrees of freedom
components are independent standard normally distributed	components are independent, scaled	components are correlated

where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

Adaptation of Sample Distribution Parameters

Adaptation: What do we want to achieve?

New search points are sampled normally distributed

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

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

- the mean vector should represent the favorite solution
- the step-size controls the step-length and thus convergence rate

  should allow to reach fastest convergence rate possible
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the shape of the distribution ellipsoid

  adaptation should allow to learn the "topography" of the problem
  particulary important for ill-conditionned problems
  $\mathbf{C} \propto H^{-1}$ on convex quadratic functions

from [Auger, p. 16]

Adaptation of the Mean

Evolution Strategies

Terminology

μ: # of parents, λ: # of offspring

Plus (elitist) and comma (non-elitist) selection

$(\mu + \lambda)$-ES: selection in {parents} $\cup$ {offspring}
(μ, λ)-ES: selection in {offspring}

$(1 + 1)$-ES

Sample one offspring from parent $\boldsymbol{m}$

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

If $\boldsymbol{x}$ better than $\boldsymbol{m}$ select

$$\boldsymbol{m} \leftarrow \boldsymbol{x}$$

why?

from [Hansen, p. 35]

Non-Elitism and Weighted Recombination

The $(\mu/\mu, \lambda)$-ES

Non-elitist selection and intermediate (weighted) recombination

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

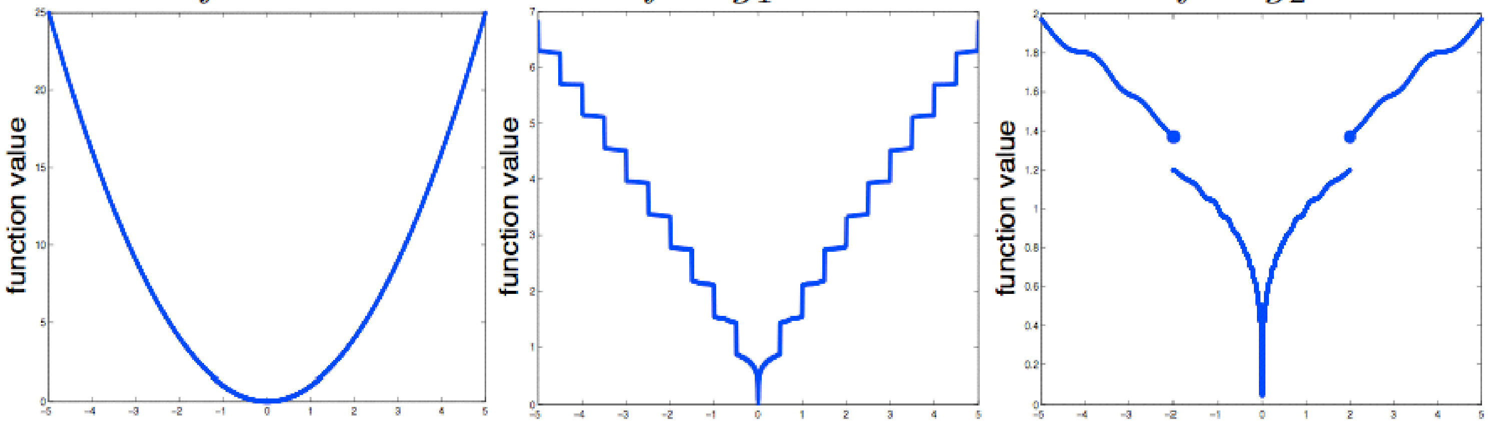
$$w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1, \quad \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} =: \mu_w \approx \frac{\lambda}{4}$$

The best μ points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.

from [Hansen, p. 34]

Invariance: Function-Value Free Property

$f = h$ $\quad f = g_1 \circ h$ $\quad f = g_2 \circ h$



Three functions belonging to the same equivalence class

A *function-value free search algorithm* is invariant under the transformation with any order preserving (strictly increasing) g.

Invariances make

- observations meaningful
- algorithms predictable and/or "robust"

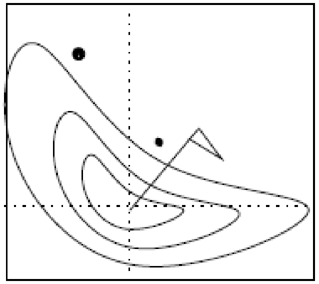
as a rigorous notion of generalization

from [Hansen, p. 37]

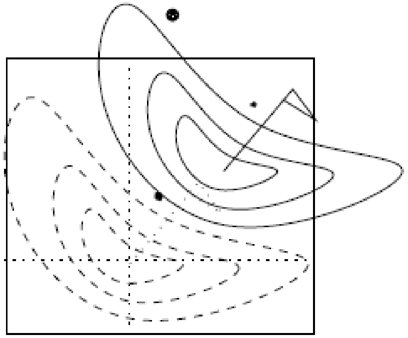
Basic Invariance in Search Space

- translation invariance

is true for most optimization algorithms



$f(\boldsymbol{x}) \leftrightarrow f(\boldsymbol{x}-\boldsymbol{a})$



Identical behavior on f and f_a

$$\begin{aligned} f: \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0 \\ f_a: \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x}-\boldsymbol{a}), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0 + \boldsymbol{a} \end{aligned}$$

No difference can be observed w.r.t. the argument of f

from [Hansen, p. 38]

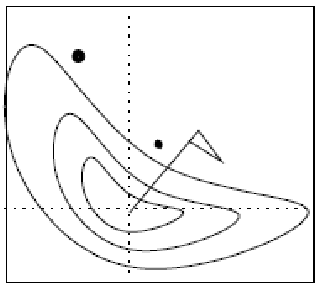
Invariance Against Search Space Rotations

Rotational Invariance in Search Space

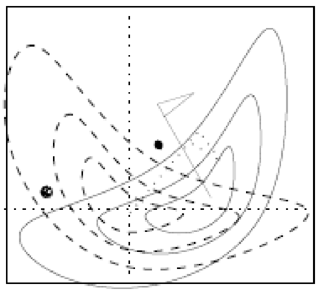
- invariance to orthogonal (rigid) transformations $\mathbf{R}$, where $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$

e.g. true for simple evolution strategies

recombination operators might jeopardize rotational invariance



$f(\boldsymbol{x}) \leftrightarrow f(\mathbf{R}\boldsymbol{x})$



Identical behavior on f and $f_{\mathbf{R}}$

$$
\begin{array}{llll}
f: & \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0 \\
f_{\mathbf{R}}: & \boldsymbol{x} \mapsto f(\mathbf{R}\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} = \mathbf{R}^{-1}(\boldsymbol{x}_0)
\end{array}
$$

45

No difference can be observed w.r.t. the argument of f

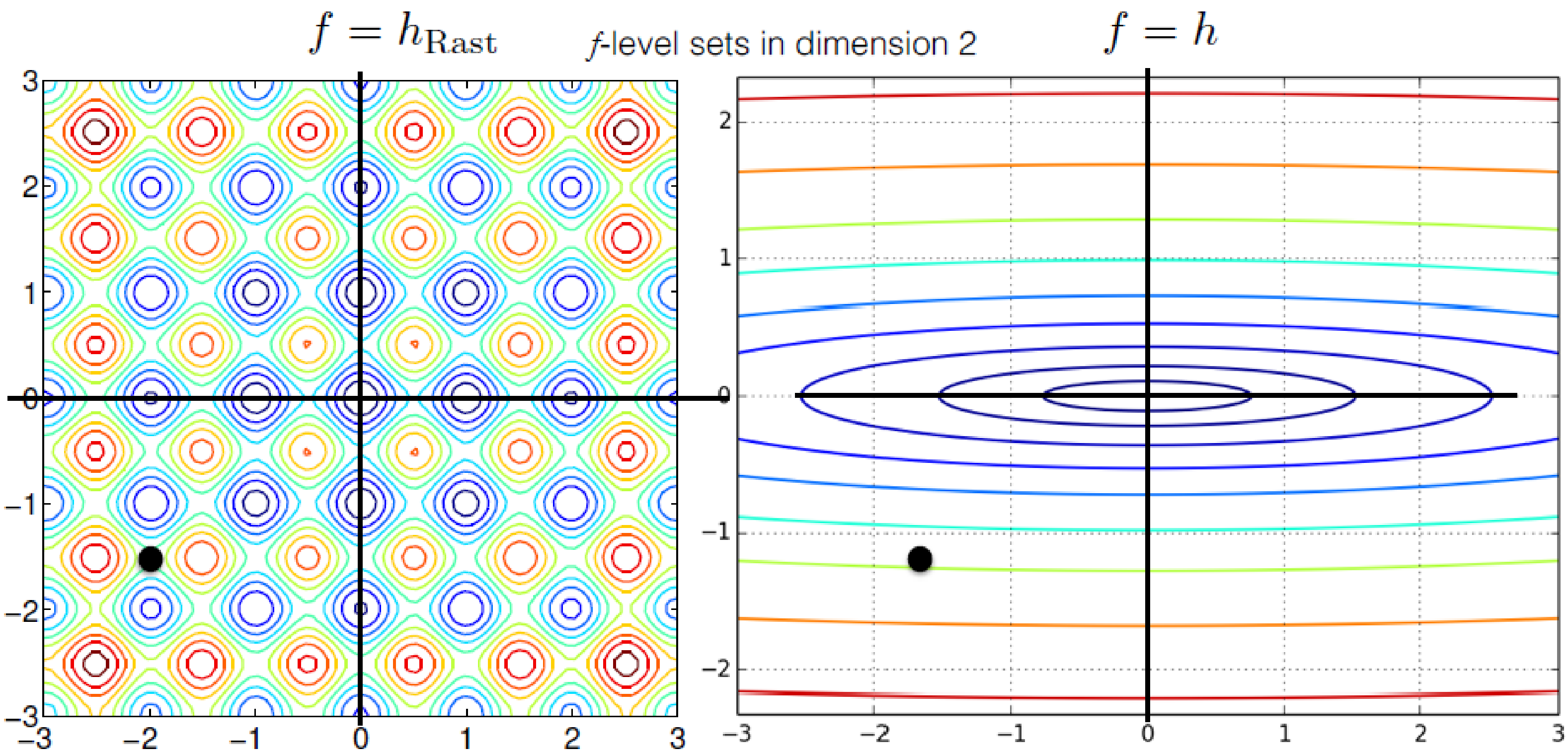
[4] Salomon 1996. "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

[5] Hansen 2000. Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies. *Parallel Problem Solving from Nature PPSN VI*

from [Hansen, p. 39]

29 / 81

Invariance Under Rigid Search Space Transformations

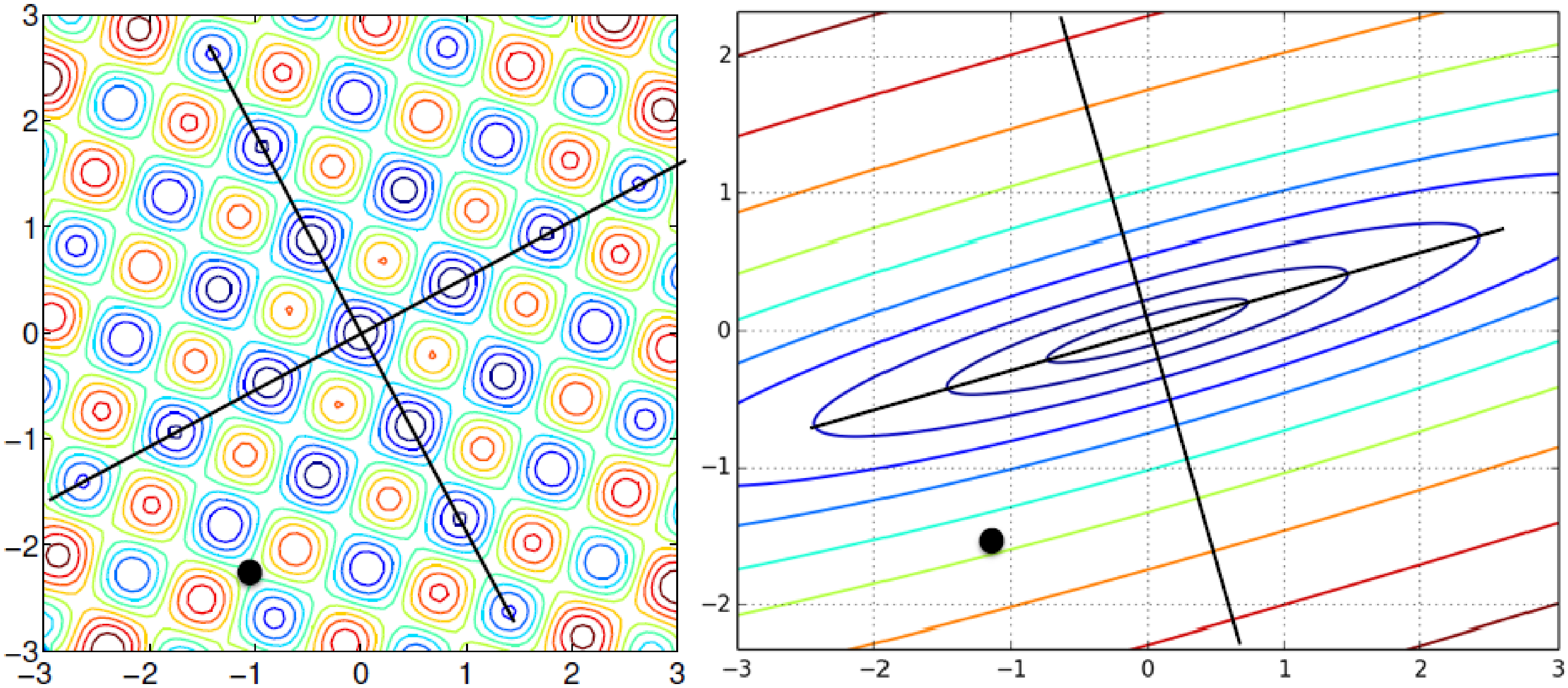


for example, invariance under search space rotation
(separable ⇔ non-separable)

from [Hansen, p. 40]

Invariance Under Rigid Search Space Transformations

$f = h_{\text{Rast}} \circ R$ f-level sets in dimension 2 $f = h \circ R$

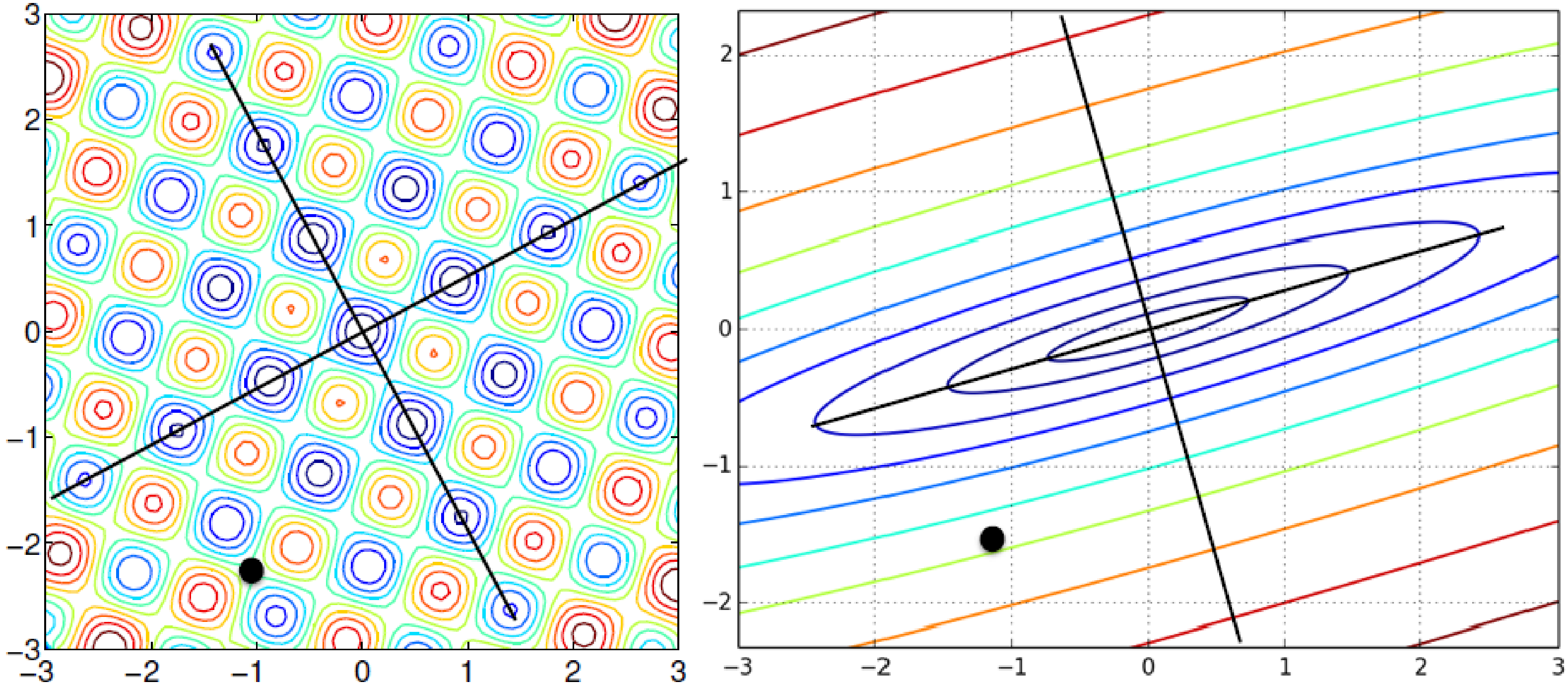


for example, invariance under search space rotation
(separable $\Leftrightarrow$ non-separable)

from [Hansen, p. 41]

Invariance Under Rigid Search Space Transformations

$f = h_{\mathrm{Rast}} \circ R$ f-level sets in dimension 2 $f = h \circ R$



for example, invariance un
(separable $\Leftrightarrow$ non-separab

mainly Nelder-Mead and CMA-ES have this property

Invariances: Summary

Invariance

The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms.
— Albert Einstein

- Empirical performance results
  - from benchmark functions
  - from solved real world problems

  are only useful if they do generalize to other problems

- Invariance is a strong non-empirical statement about generalization

  generalizing (identical) performance from a single function to a whole class of functions

consequently, invariance is important for the evaluation of search algorithms

from [Hansen, p. 43]

Step-Size Adaptation

Recap CMA-ES: What We Have So Far

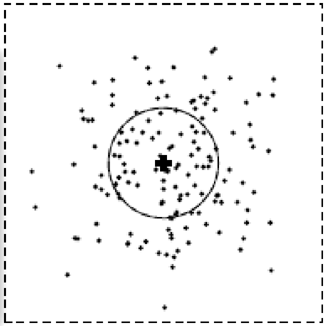
Evolution Strategies

Recalling

New search points are sampled normally distributed

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

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



where

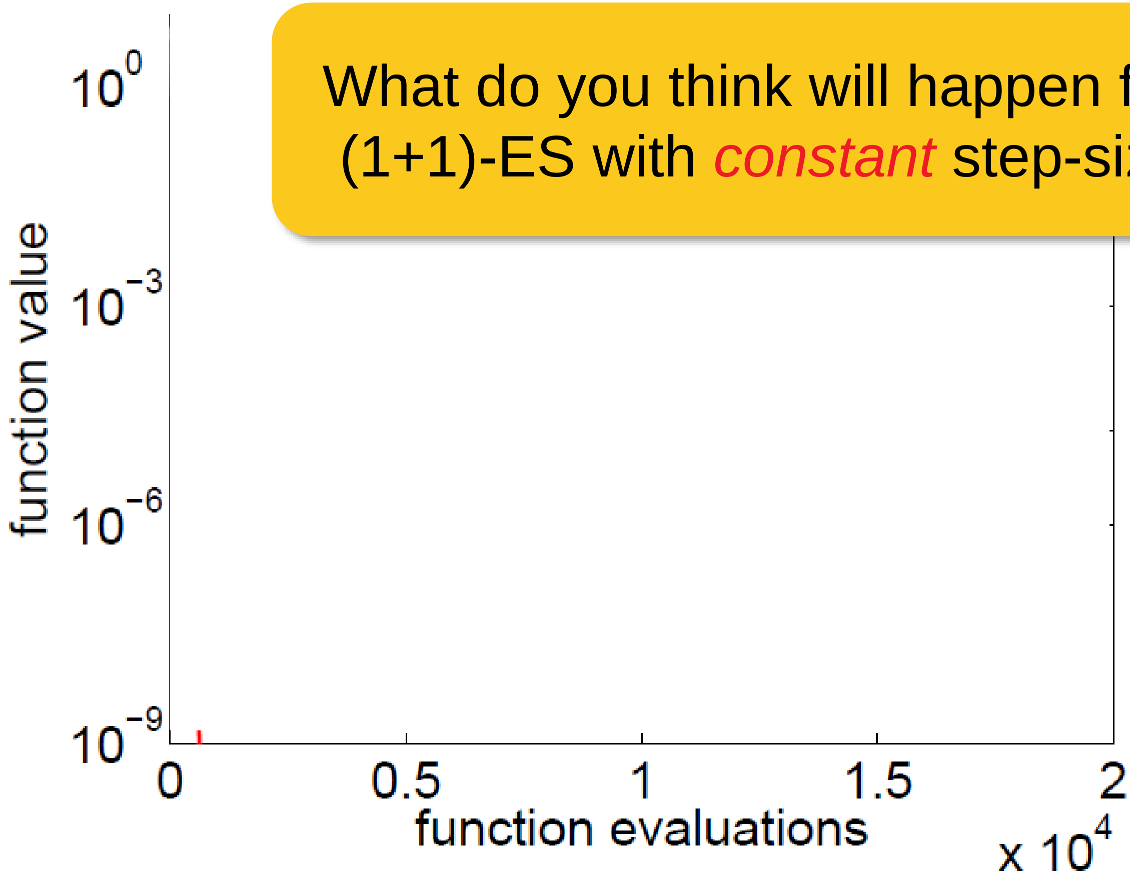
- the mean vector $m \in \mathbb{R}^n$ represents the favorite solution and $m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda}$
- the so-called step-size $\sigma \in \mathbb{R}_+$ controls the *step length*
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the shape of the distribution ellipsoid

The remaining question is how to update σ and $\mathbf{C}$.

from [Hansen, p. 45]

Why Step-Size Control?

What do you think will happen for a (1+1)-ES with *constant* step-size?

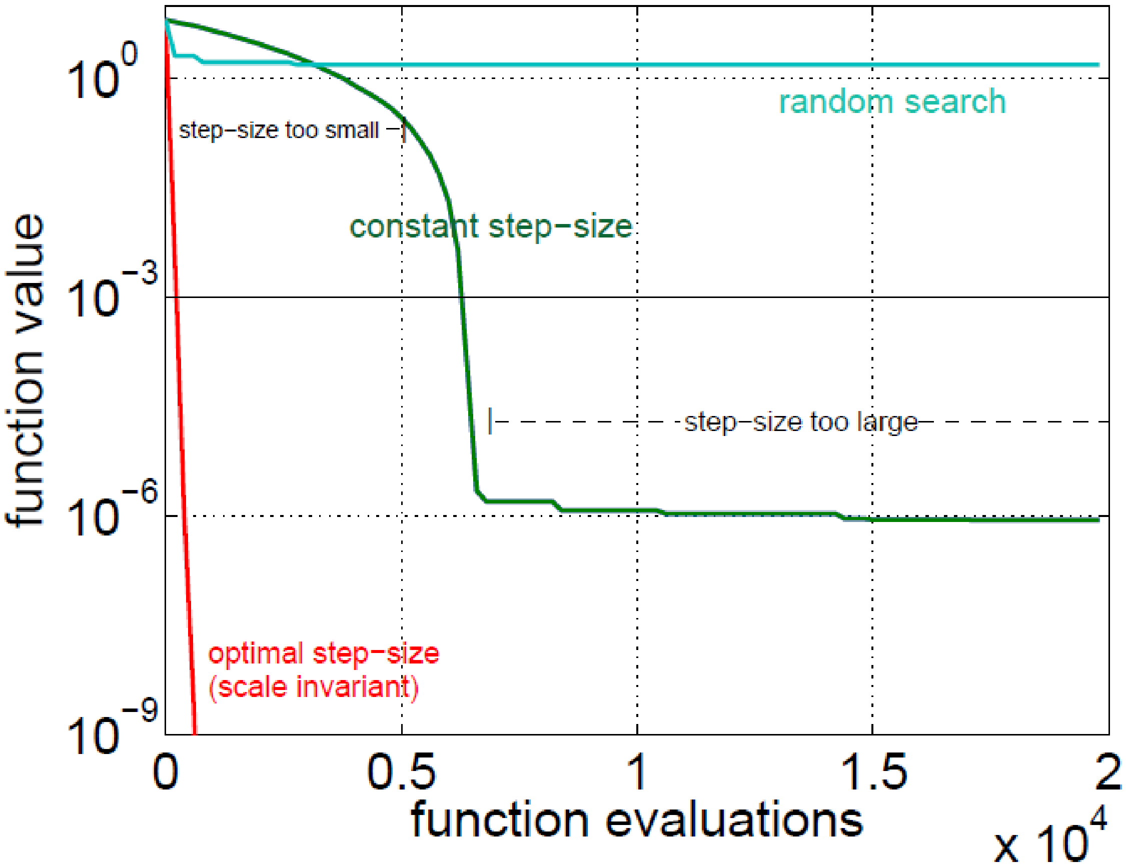


$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 10$

from [Auger, p. 22]

Why Step-Size Control?

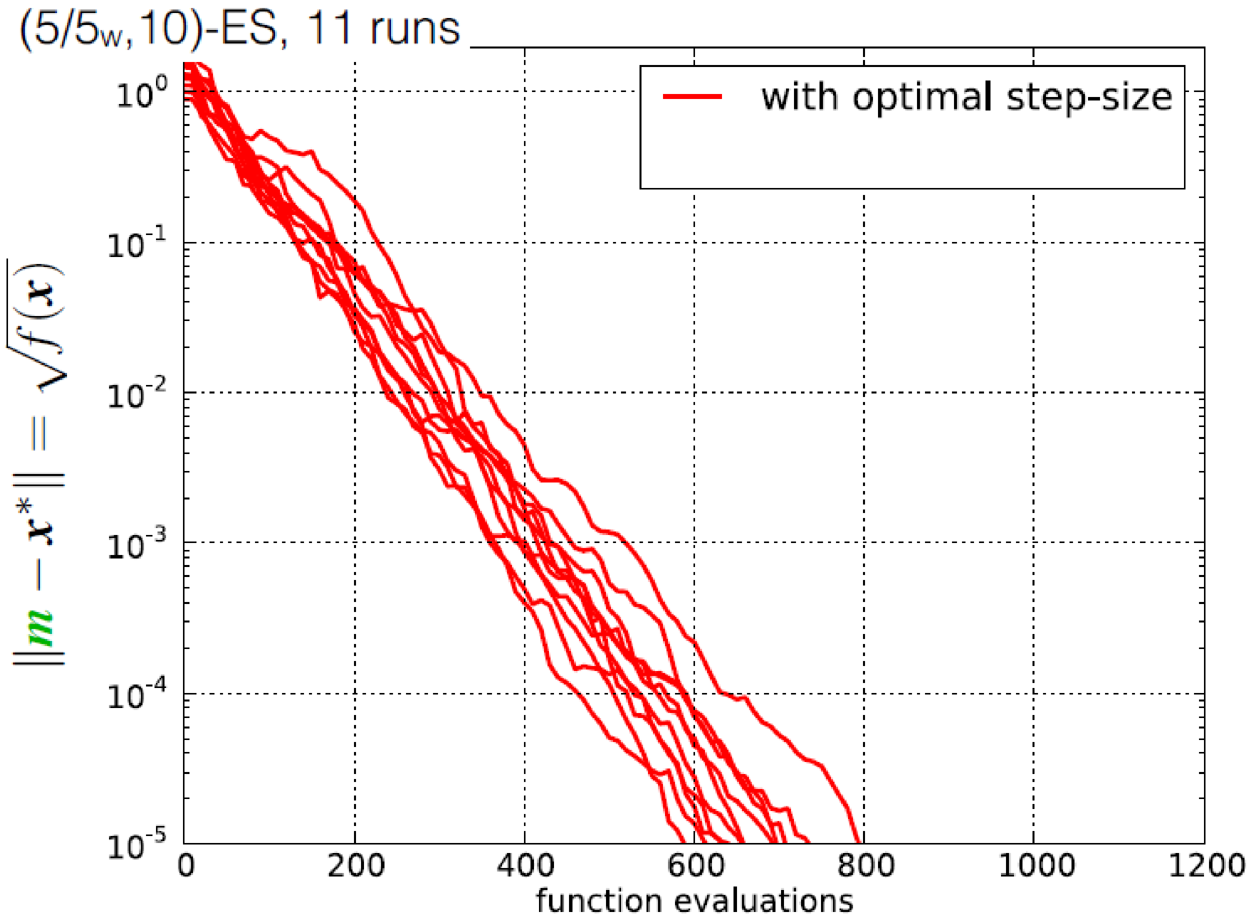


$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 10$

from [Auger, p. 22]

Why Step-Size Control?



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

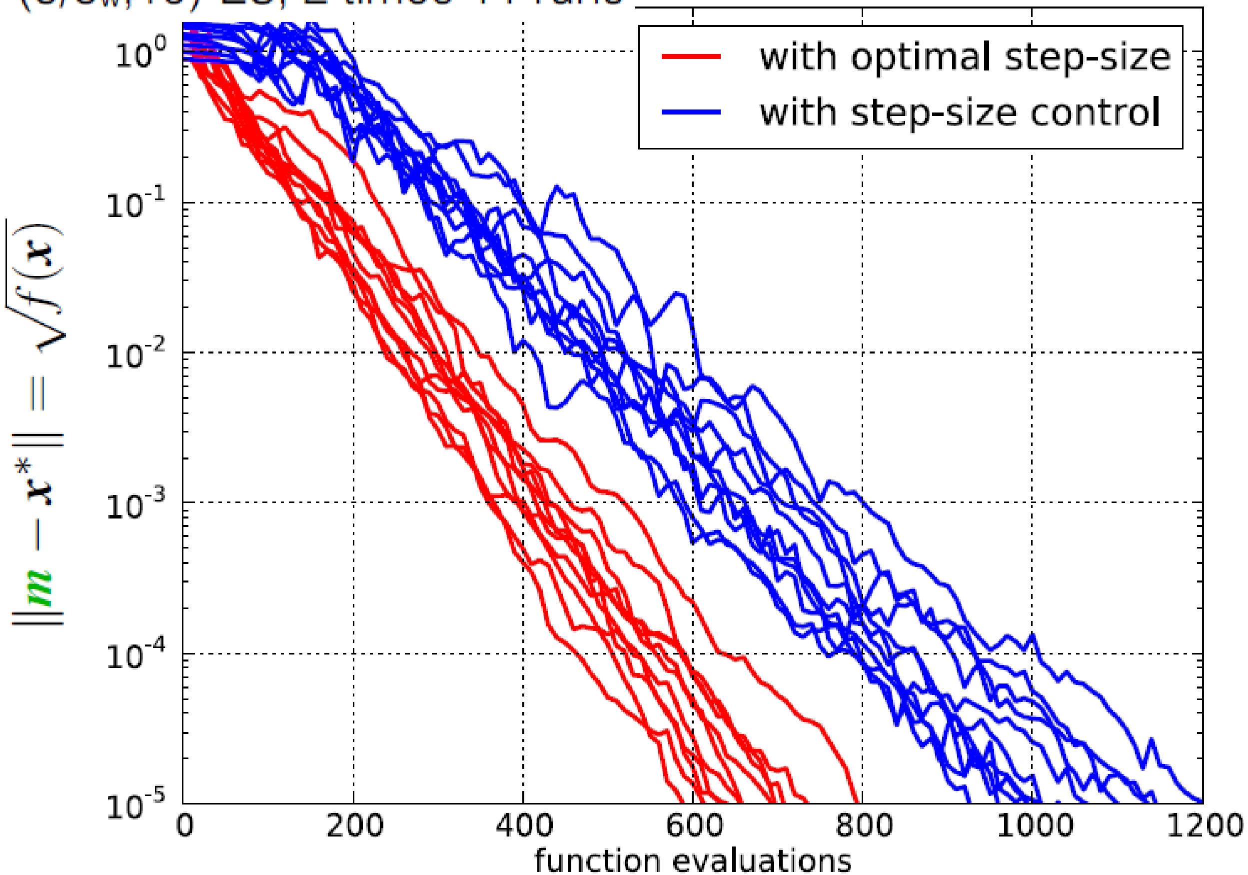
for $n = 10$ and $\boldsymbol{x}^0 \in [-0.2, 0.8]^n$

with optimal step-size σ

from [Hansen, p. 47]

Why Step-Size Control?

(5/5w,10)-ES, 2 times 11 runs



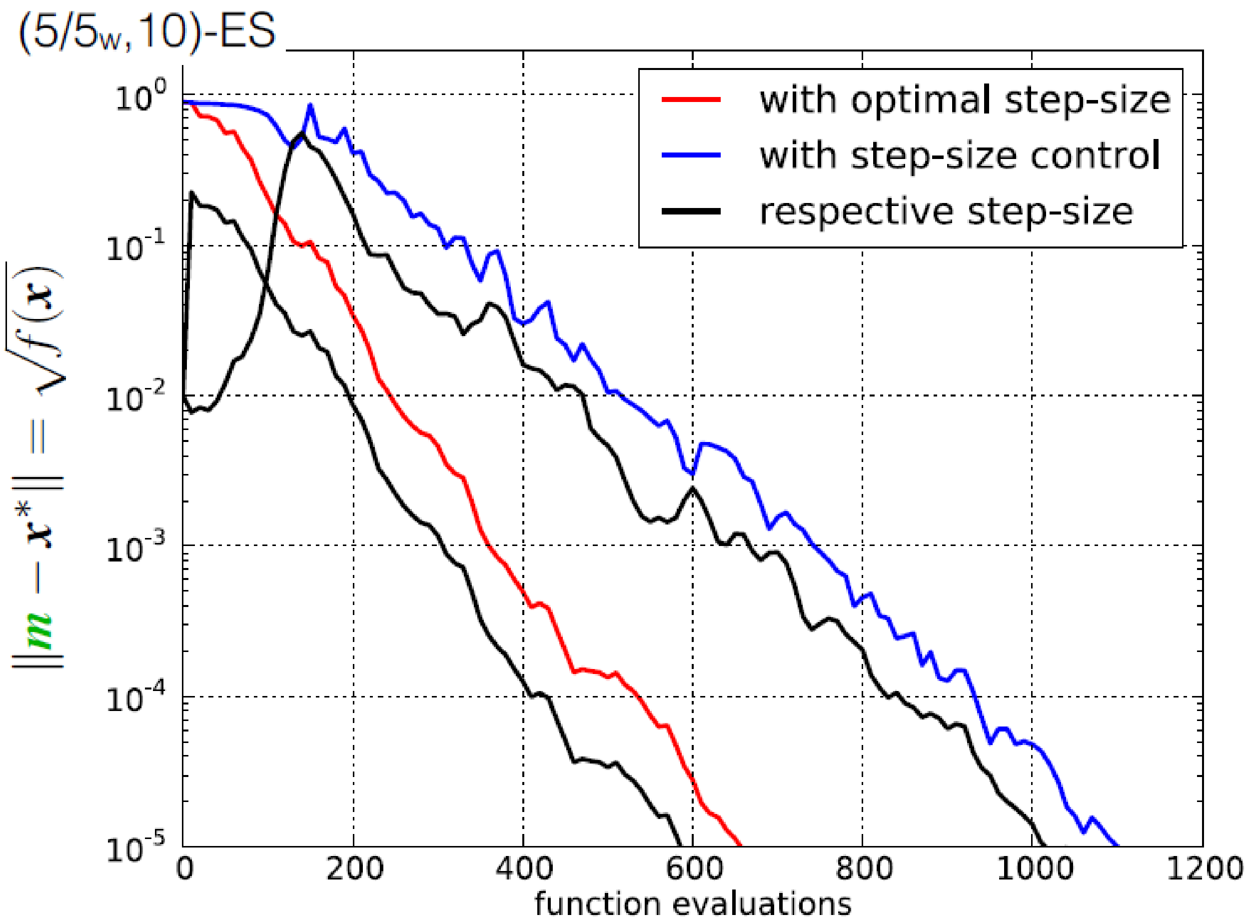
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

for $n = 10$ and $\boldsymbol{x}^0 \in [-0.2, 0.8]^n$

with optimal versus adaptive step-size σ with too small initial σ

from [Hansen, p. 48]

Why Step-Size Control?



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

for $n = 10$ and $\boldsymbol{x}^0 \in [-0.2, 0.8]^n$

comparing number of f-evals to reach $\|\boldsymbol{m}\| = 10^{-5}$: $\frac{1100-100}{650} \approx 1.5$

from [Hansen, p. 49]

Adapting the Step-Size

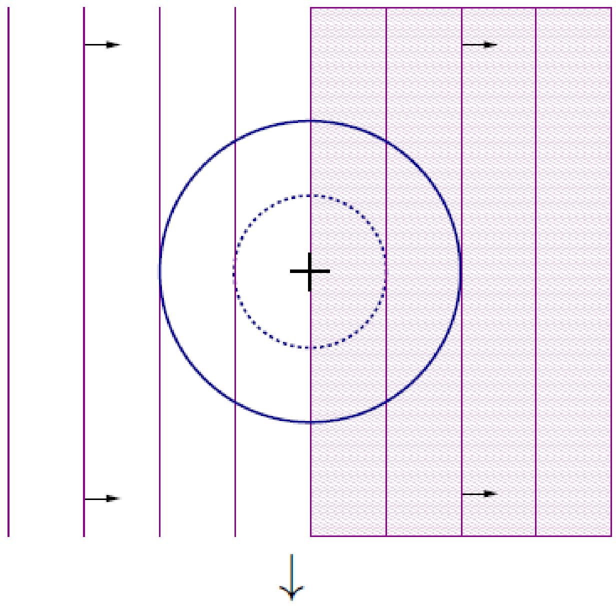
Question:

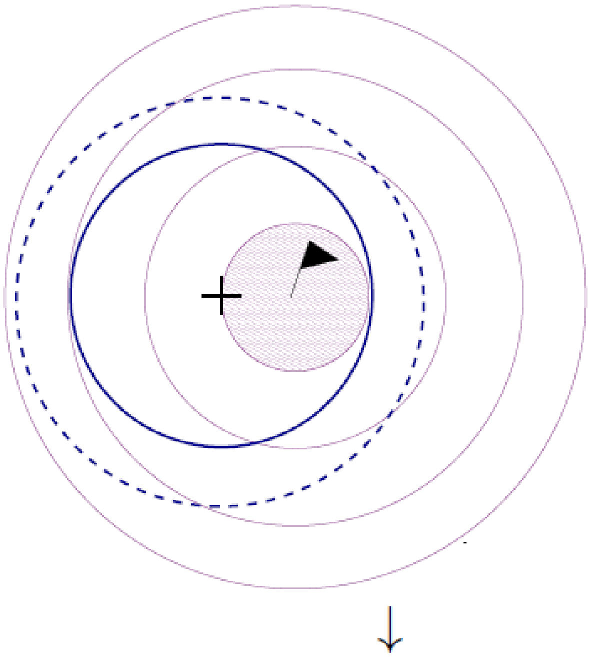
How to actually adapt the step-size during the optimization?

Most common:

- 1/5 success rule
- Cumulative Step-Size Adaptation (CSA, as in standard CMA-ES)
- others possible (Two-Point Adaptation, self-adaptive step-size, ...)

One-fifth success rule

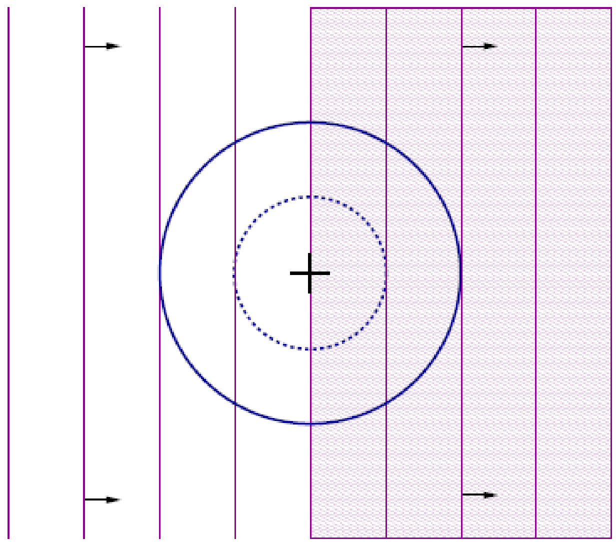


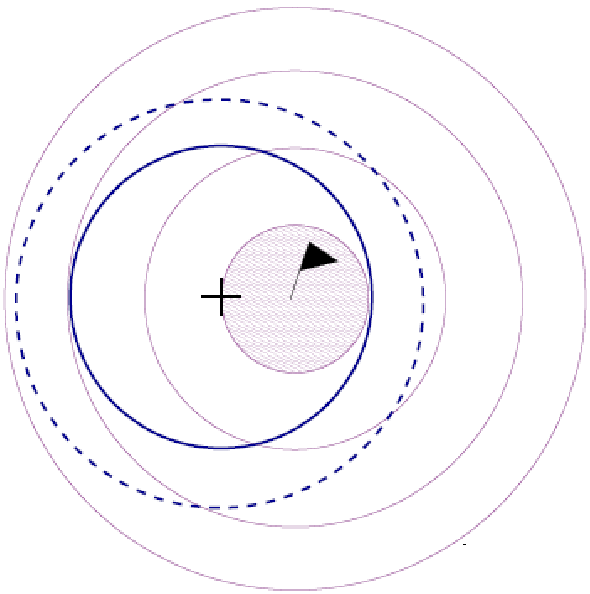


from [Auger, p. 32]

One-Fifth Success Rule

One-fifth success rule





Probability of success (p_s)

Probability of success (p_s)

$1/2$

$1/5$

"too small"

from [Auger, p. 33]

One-fifth success rule

p_s: # of successful offspring / # offspring (per generation)

$$\sigma \leftarrow \sigma \times \exp\left(\frac{1}{3} \times \frac{p_s - p_{\text{target}}}{1 - p_{\text{target}}}\right)$$

Increase σ if $p_s > p_{\text{target}}$
Decrease σ if $p_s < p_{\text{target}}$

$(1+1)$-ES

$p_{target} = 1/5$

IF *offspring better parent*

$p_s = 1$, $\sigma \leftarrow \sigma \times \exp(1/3)$

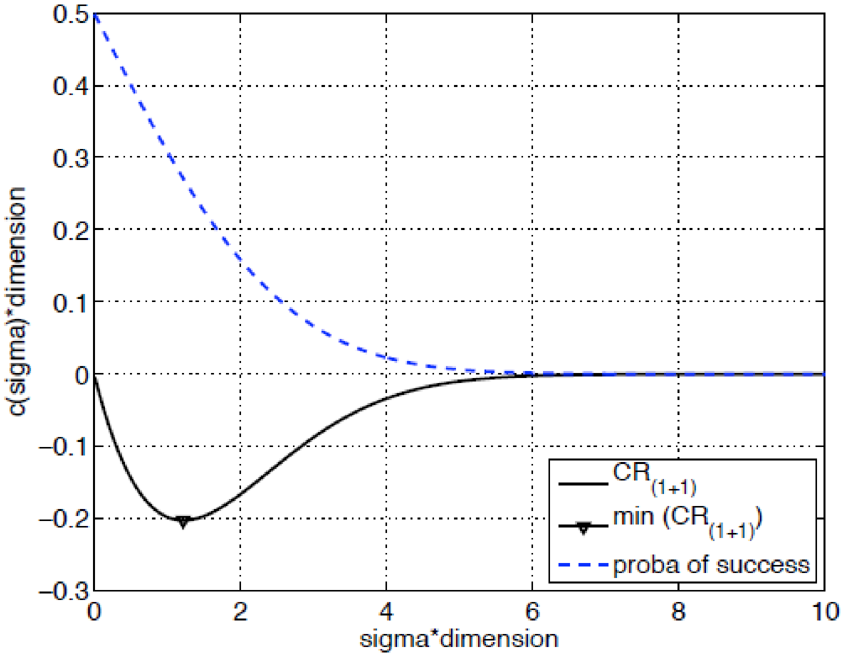
ELSE

$p_s = 0$, $\sigma \leftarrow \sigma / \exp(1/3)^{1/4}$

from [Auger, p. 34]

Why 1/5?

Asymptotic convergence rate and probability of success of scale-invariant step-size (1+1)-ES



sphere - asymptotic results, i.e. $n = \infty$ (see slides before)

1/5 trade-off of optimal probability of success on the sphere and corridor

from [Auger, p. 35]

Cumulative Step-Size Adaptation (CSA)

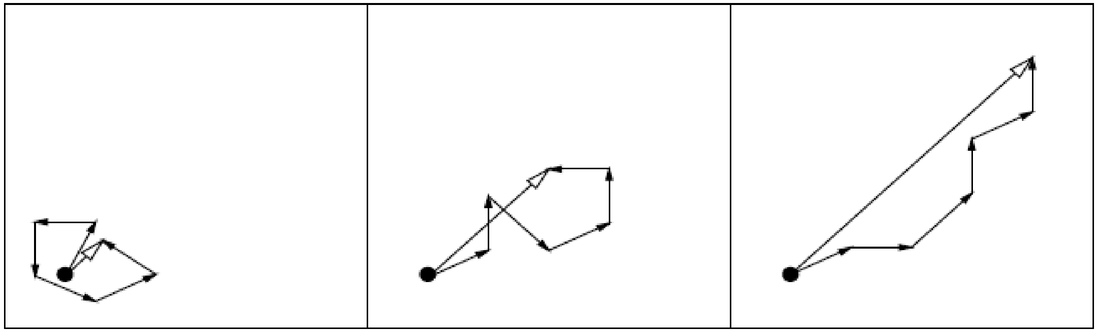
Path Length Control (CSA)

The Concept of Cumulative Step-Size Adaptation

$$\begin{aligned} \boldsymbol{x}_i &= \boldsymbol{m} + \sigma\, \boldsymbol{y}_i \\ \boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma\, \boldsymbol{y}_w \end{aligned}$$

Measure the length of the *evolution path*

the pathway of the mean vector $\boldsymbol{m}$ in the generation sequence



$\Downarrow$ decrease σ

$\Downarrow$ increase σ

from [Auger, p. 36]

Path Length Control (CSA)

The Equations

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

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

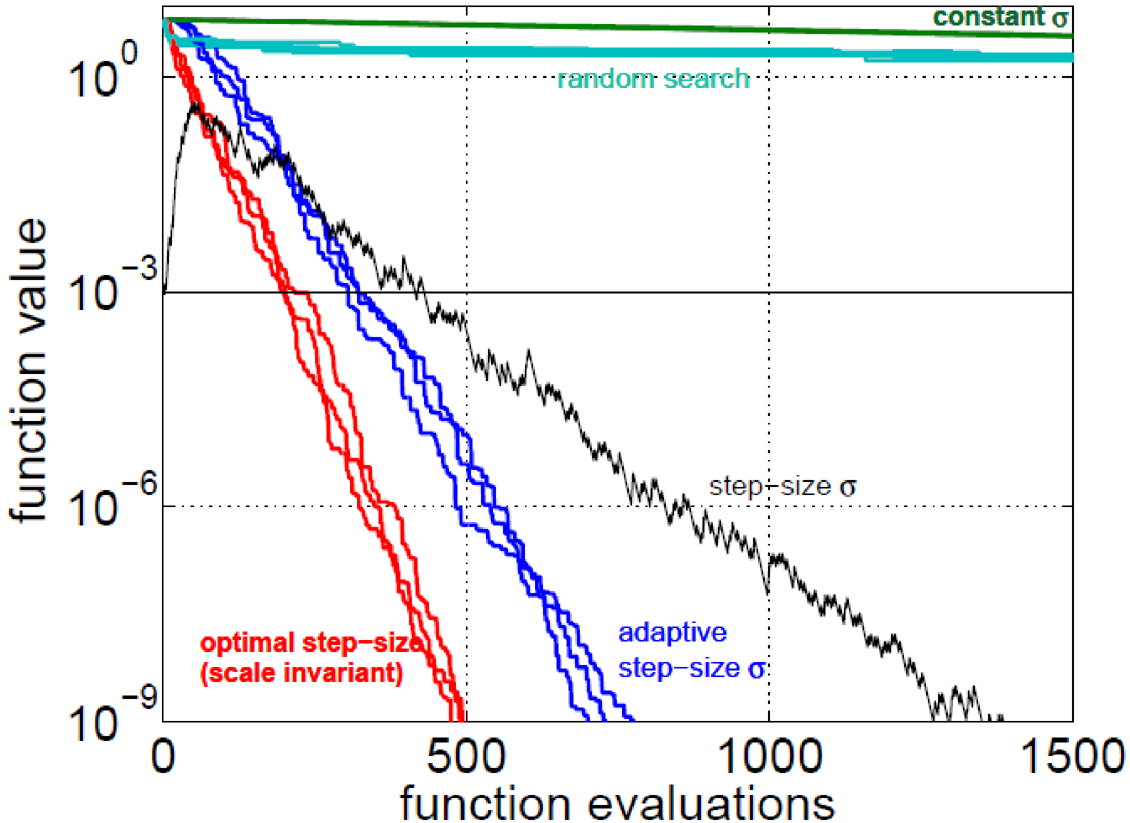
$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}}_{\text{accounts for } 1-c_\sigma} \quad \underbrace{\sqrt{\mu_w}}_{\text{accounts for } w_i} \quad \boldsymbol{y}_w$$

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

from [Auger, p. 37]

Step-size adaptation

What is achived



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

in $[-0.2, 0.8]^n$
for $n = 10$

Linear convergence

from [Auger, p. 38]

Covariance Matrix Adaptation

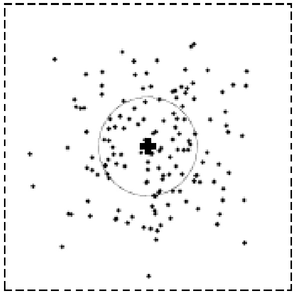
Evolution Strategies

Recalling

New search points are sampled normally distributed

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

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



where

- the mean vector $m \in \mathbb{R}^n$ 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

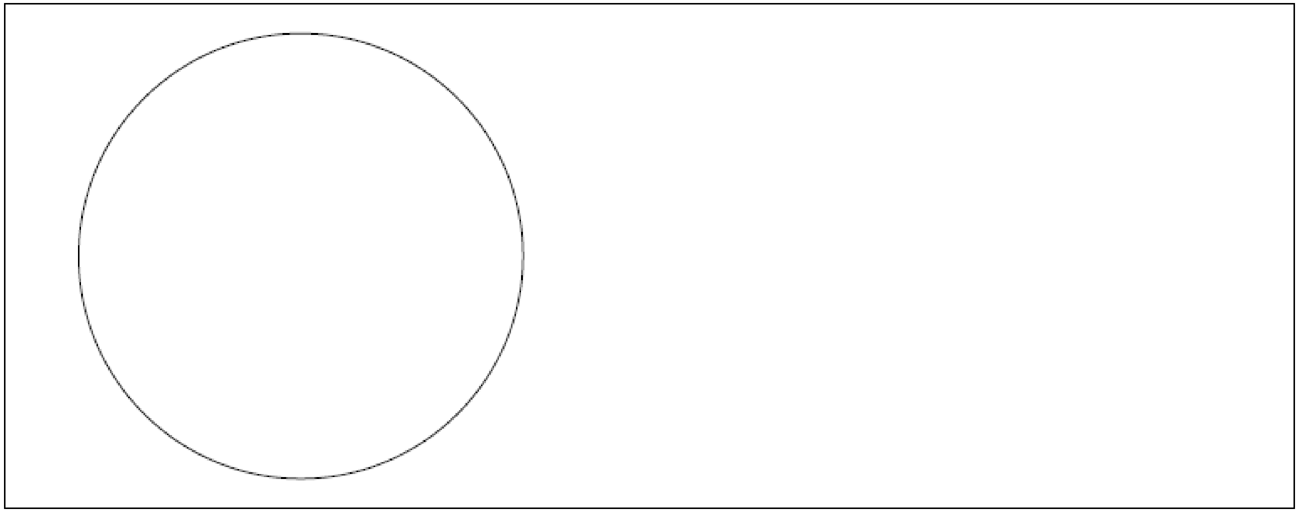
The remaining question is how to update $\mathbf{C}$.

from [Auger, p. 40]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



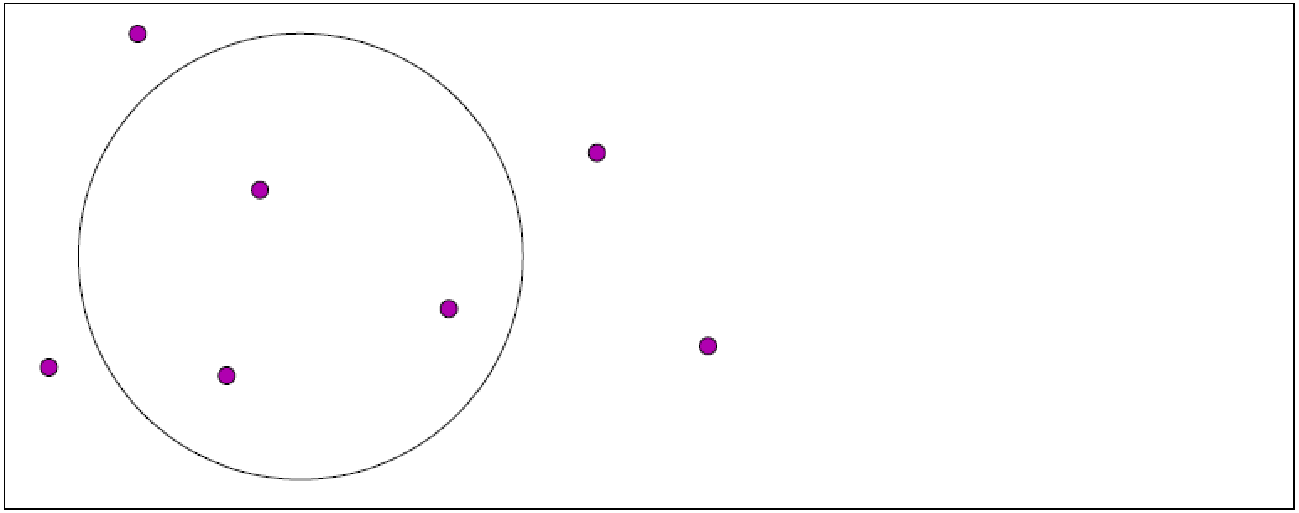
initial distribution, $\mathbf{C} = \mathbf{I}$

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



initial distribution, $\mathbf{C} = \mathbf{I}$

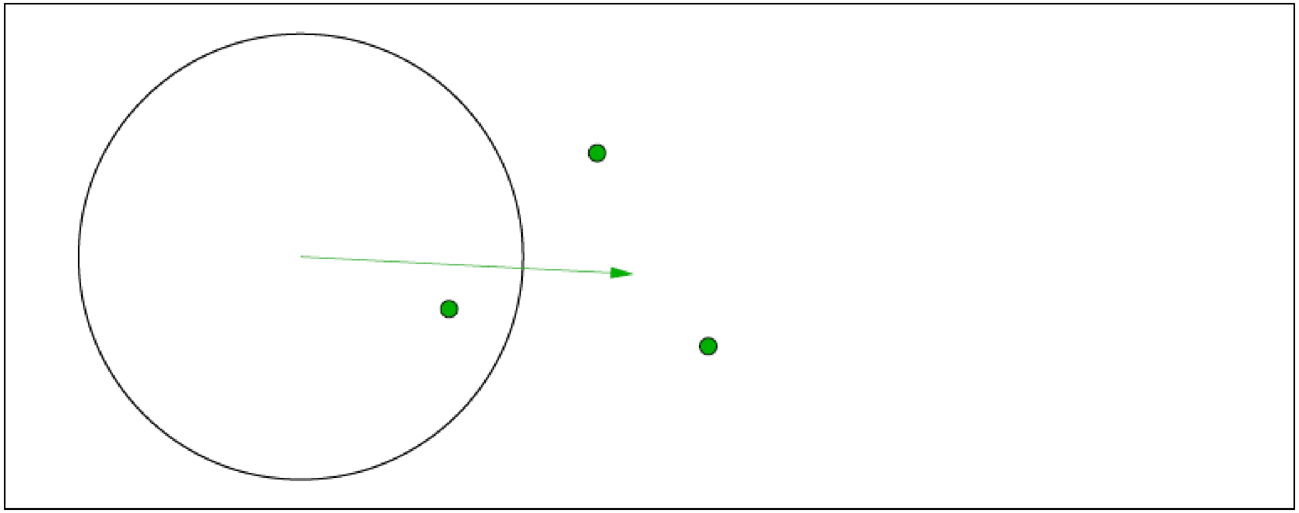
from [Auger, p. 41]

Rank-One Update of Covariance Matrix

Covariance Matrix Adaptation

Rank-One Update

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



$\mathbf{y}_w$, movement of the population mean $\mathbf{m}$ (disregarding σ)

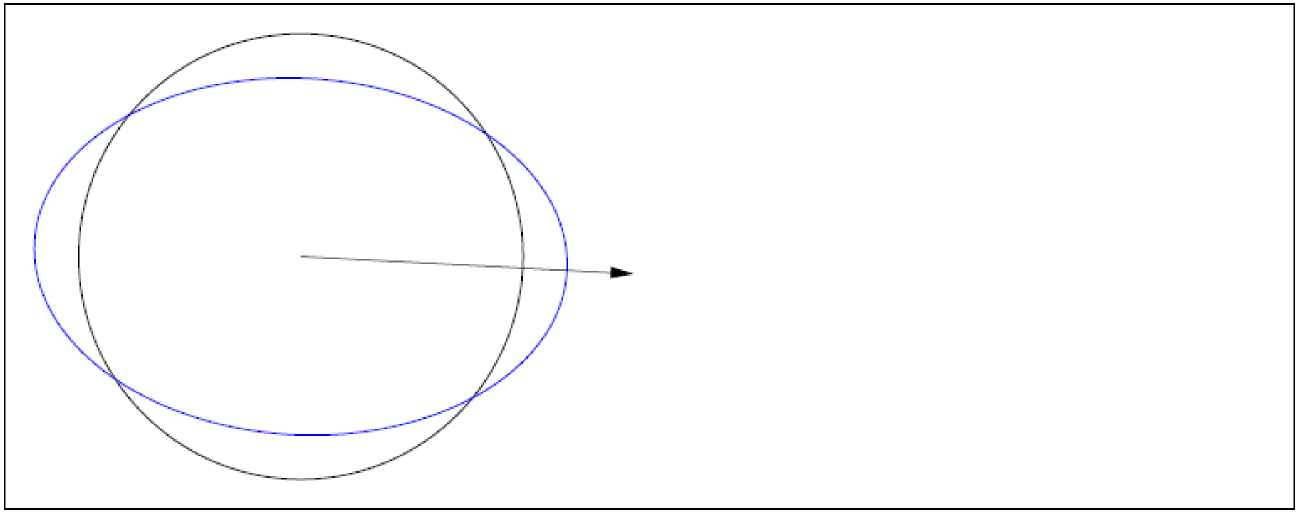
from [Auger, p. 41]

Rank-One Update of Covariance Matrix

Covariance Matrix Adaptation

Rank-One Update

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



mixture of distribution $\mathbf{C}$ and step $\boldsymbol{y}_w$,

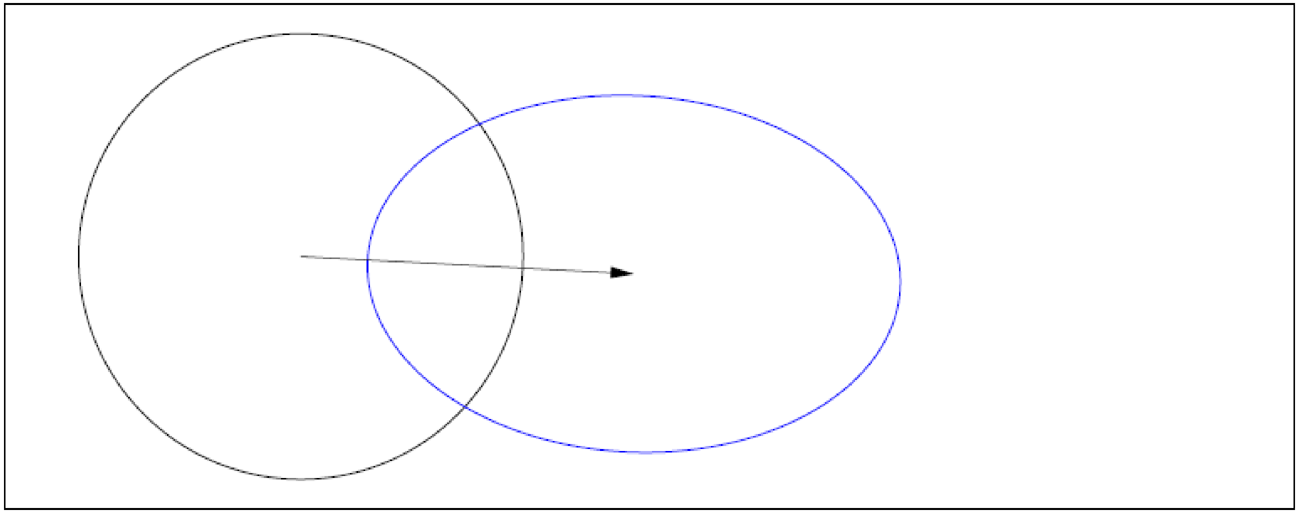
$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



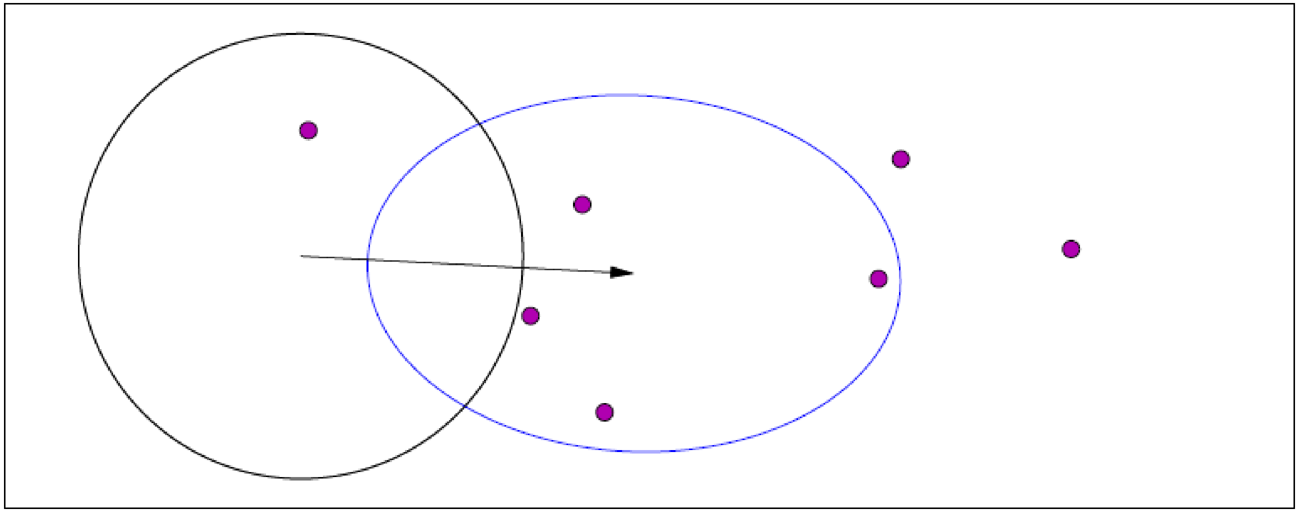
new distribution (disregarding σ)

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

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



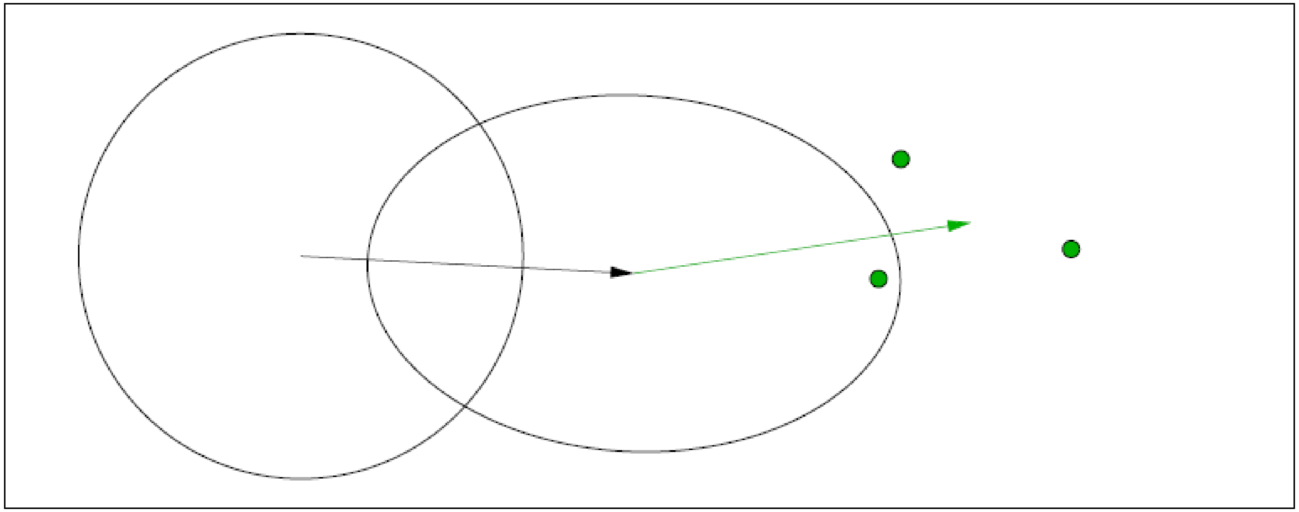
new distribution (disregarding σ)

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



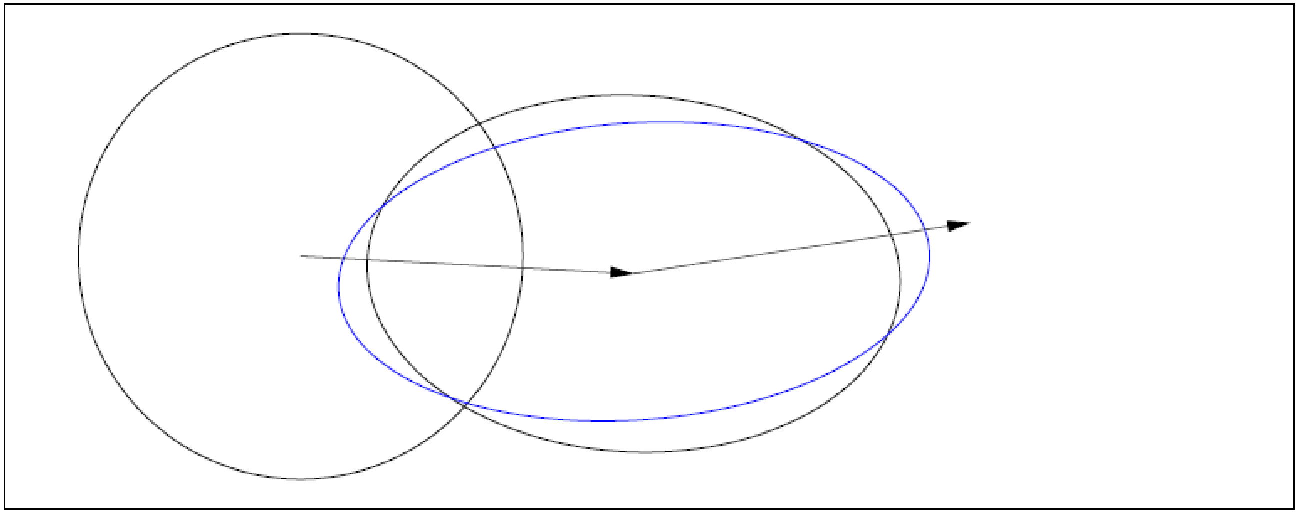
movement of the population mean m

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



mixture of distribution $\mathbf{C}$ and step y_w,

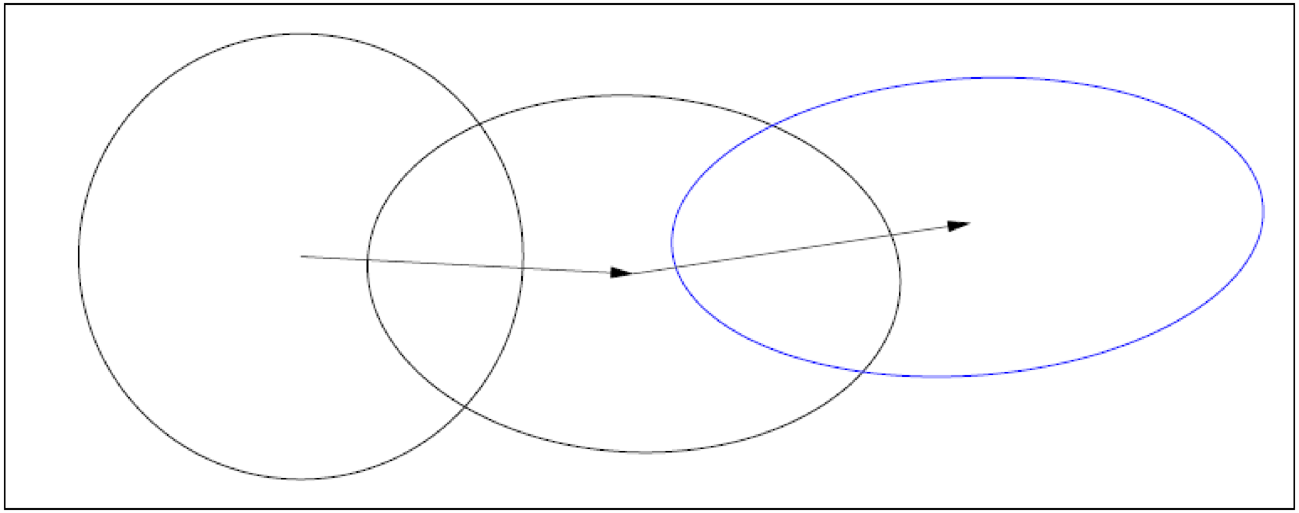
$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times y_w y_w^{\mathrm{T}}$

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

$$m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i \, y_{i:\lambda}, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



new distribution,

$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times y_w y_w^{\mathrm{T}}$

the ruling principle: the adaptation increases the likelihood of successful steps, y_w, to appear again

from [Auger, p. 41]

Covariance Matrix Adaptation

Rank-One Update

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

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

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

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

from [Auger, p. 42]

Rank-One Update: Summary

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

covariance matrix adaptation

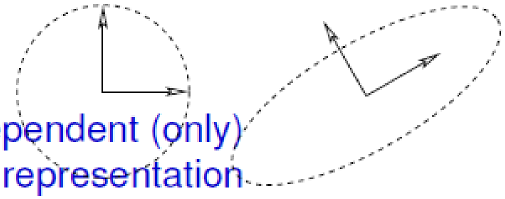
- learns all pairwise dependencies between variables

  off-diagonal entries in the covariance matrix reflect the dependencies

- conducts a principle component analysis (PCA) of steps $\boldsymbol{y}_w$, sequentially in time and space

  eigenvectors of the covariance matrix $\mathbf{C}$ 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

- learns a new (Mahalanobis) metric

  variable metric method

- approximates the inverse Hessian on quadratic functions

  transformation into the sphere function

- for $\mu = 1$: conducts a natural gradient ascent on the distribution $\mathcal{N}$

  entirely independent of the given coordinate system

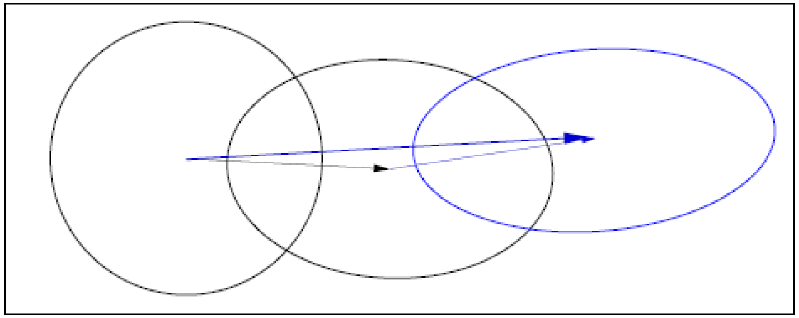
from [Hansen, p. 71]

Cumulation

The Evolution Path

Evolution Path

Conceptually, the evolution path is 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 $\boldsymbol{m}$.



An exponentially weighted sum of steps $\boldsymbol{y}_w$ is used

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

The recursive construction of the evolution path (cumulation):

$$p_c \leftarrow \underbrace{(1-c_c)}_{\text{decay factor}} p_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu_w}}_{\text{normalization factor}} \underbrace{\boldsymbol{y}_w}_{\text{input } = \frac{m - m_{\text{old}}}{\sigma}}$$

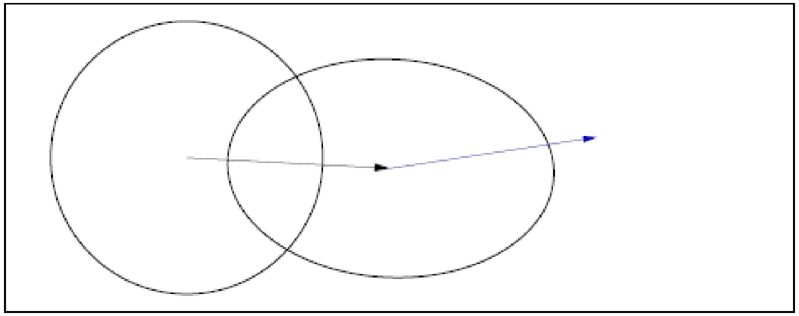
where $\mu_w = \frac{1}{\sum {w_i}^2}$, $c_c \ll 1$. History information is accumulated in the evolution path.

from [Auger, p. 44]

Cumulation

Utilizing the Evolution Path

We used $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$ for updating $\mathbf{C}$. Because $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}} = -\boldsymbol{y}_w(-\boldsymbol{y}_w)^{\mathrm{T}}$ the sign of $\boldsymbol{y}_w$ is lost.

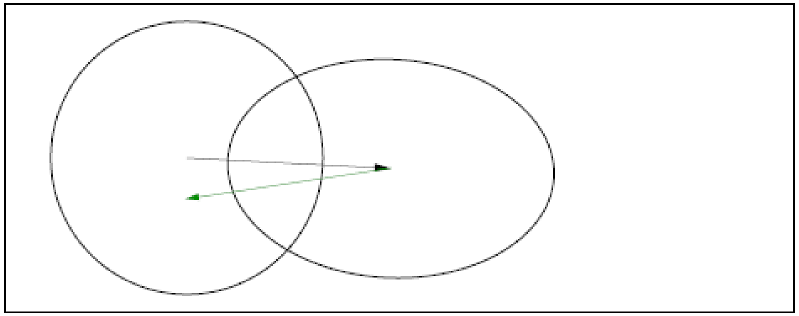


from [Auger, p. 45]

Cumulation

Utilizing the Evolution Path

We used $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$ for updating **C**. Because $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}} = -\boldsymbol{y}_w(-\boldsymbol{y}_w)^{\mathrm{T}}$ the sign of $\boldsymbol{y}_w$ is lost.

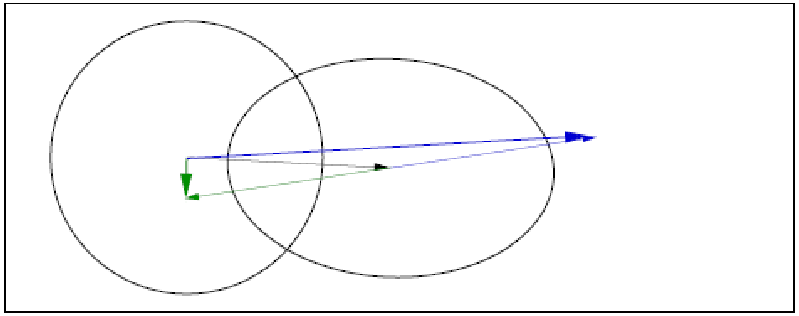


from [Auger, p. 45

Cumulation

Utilizing the Evolution Path

We used $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$ for updating $\mathbf{C}$. Because $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}} = -\boldsymbol{y}_w(-\boldsymbol{y}_w)^{\mathrm{T}}$ the sign of $\boldsymbol{y}_w$ is lost.



The sign information is (re-)introduced by using the *evolution path*.

$$p_c \leftarrow \underbrace{(1 - c_c)}_{\text{decay factor}} p_c + \underbrace{\sqrt{1 - (1 - c_c)^2}\sqrt{\mu_w}}_{\text{normalization factor}} \boldsymbol{y}_w$$

$$\mathbf{C} \leftarrow (1 - c_{\text{cov}})\mathbf{C} + c_{\text{cov}} \underbrace{p_c p_c{}^{\mathrm{T}}}_{\text{rank-one}}$$

where $\mu_w = \frac{1}{\sum w_i{}^2}$, $c_c \ll 1$.

from [Auger, p. 45]

Rank-μ Update

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

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.

from [Auger, p. 47]

Rank-μ Update

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

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 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.

from [Auger, p. 47]

Rank-μ Update

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

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 matrix

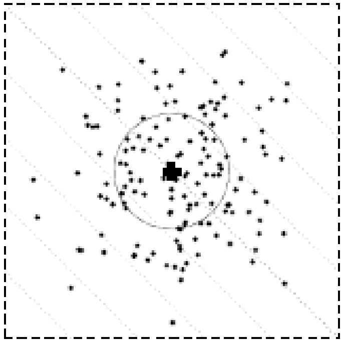
$$\mathbf{C}_\mu = \sum_{i=1}^{\mu} w_i\, \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^{\mathrm{T}}$$

computes a weighted mean of the outer products of the best μ steps and has rank $\min(\mu, n)$ with probability one.
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 $c_{\mathrm{cov}} \leq 1$.

from [Auger, p. 47]

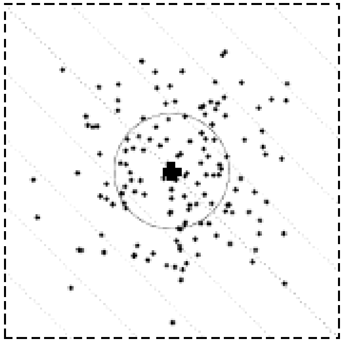


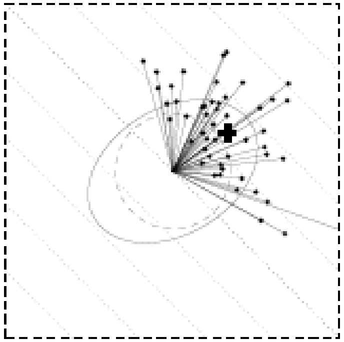
$$x_i \quad = \quad m + \sigma\, y_i, \quad y_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$$

sampling of
$\lambda = 150$ solutions
where $\mathbf{C} = \mathbf{I}$ and
$\sigma = 1$

from [Auger, p. 48]

Illustration of Rank-μ Update





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

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

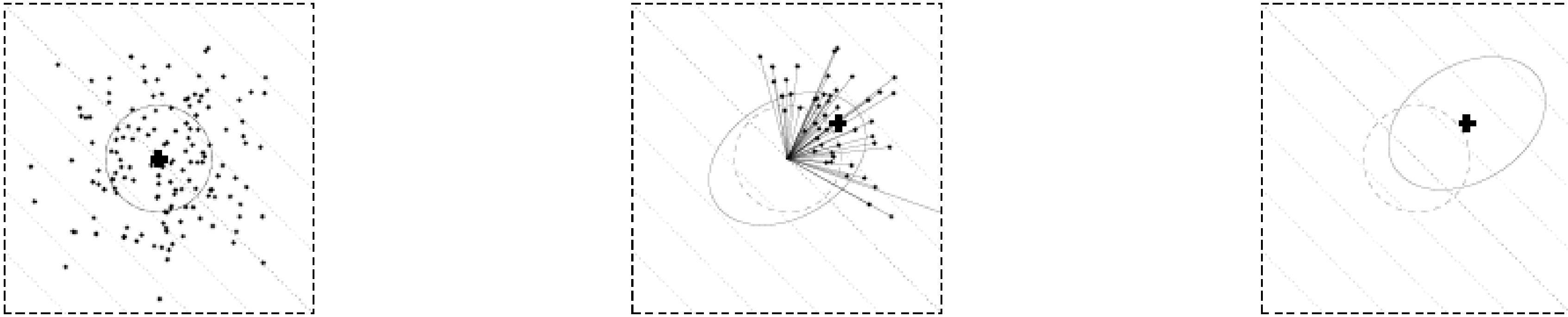
$$\mathbf{C} \leftarrow (1 - 1) \times \mathbf{C} + 1 \times \mathbf{C}_\mu$$

sampling of $\lambda = 150$ solutions where $\mathbf{C} = \mathbf{I}$ and $\sigma = 1$

calculating $\mathbf{C}$ where $\mu = 50$, $w_1 = \cdots = w_\mu = \frac{1}{\mu}$, and $c_{\mathrm{cov}} = 1$

from [Auger, p. 48]

Illustration of Rank-μ Update



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

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

$\mathbf{C} \leftarrow (1-1)\times \mathbf{C} + 1 \times \mathbf{C}_\mu$

$m_{\mathrm{new}} \leftarrow m + \frac{1}{\mu}\sum \mathbf{y}_{i:\lambda}$

sampling of $\lambda = 150$ solutions where $\mathbf{C} = \mathbf{I}$ and $\sigma = 1$

calculating $\mathbf{C}$ where $\mu = 50$, $w_1 = \cdots = w_\mu = \frac{1}{\mu}$, and $c_{\mathrm{cov}} = 1$

new distribution

from [Auger, p. 48]

Rank-μ Update: Summary

The rank-μ update

- increases the possible learning rate for large populations

"large" when $\lambda \geq 3n + 10$

- is the primary mechanism whenever a large population size is used
- can be easily combined with rank-one update

The CMA-ES

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $p_c = \mathbf{0}$,
Set: $c_c \approx 4/n$, $c_\sigma \approx 4/n$, c_1 ... $\frac{\mu_w}{n}$,
and $w_{i=1\ldots\lambda}$ such that $\mu_w =$

Promised:
Understand the main principles of this state-of-the-art algorithm.

While not terminate

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

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda} = m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda} \qquad \text{update mean}$$

$$p_c \leftarrow (1 - c_c) p_c + \mathbb{1}_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, y_w \qquad \text{cumulation for } \mathbf{C}$$

$$p_\sigma \leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \mathbf{C}^{-\frac{1}{2}} y_w \qquad \text{cumulation for } \sigma$$

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu) \mathbf{C} + c_1 p_c p_c^{\mathrm{T}} + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda} y_{i:\lambda}^{\mathrm{T}} \qquad \text{update } \mathbf{C}$$

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

Not covered on this slide: termination, restarts, useful output, boundaries and encoding

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_c = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
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

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

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

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

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

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

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

Not covered on this slide: termination, restarts, useful output, boundaries and encoding

Strategy Internal Parameters

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

Parameters were identified in carefully chosen experimental set ups. Parameters do not in the first place depend on the objective function and are not meant to be in the users choice.
Only(?) the population size λ (and the initial σ) might be reasonably varied in a wide range, *depending on the objective function*
Useful: restarts with increasing population size (IPOP)

from [Hansen, p. 90]

Experimental Considerations

Experimentum Crucis (0)

What did we want to achieve?

- reduce any convex-quadratic function

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

e.g. $f(\boldsymbol{x}) = \sum_{i=1}^{n} 10^{6\frac{i-1}{n-1}} x_i^2$

to the sphere model

$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$$

without use of derivatives

- lines of equal density align with lines of equal fitness

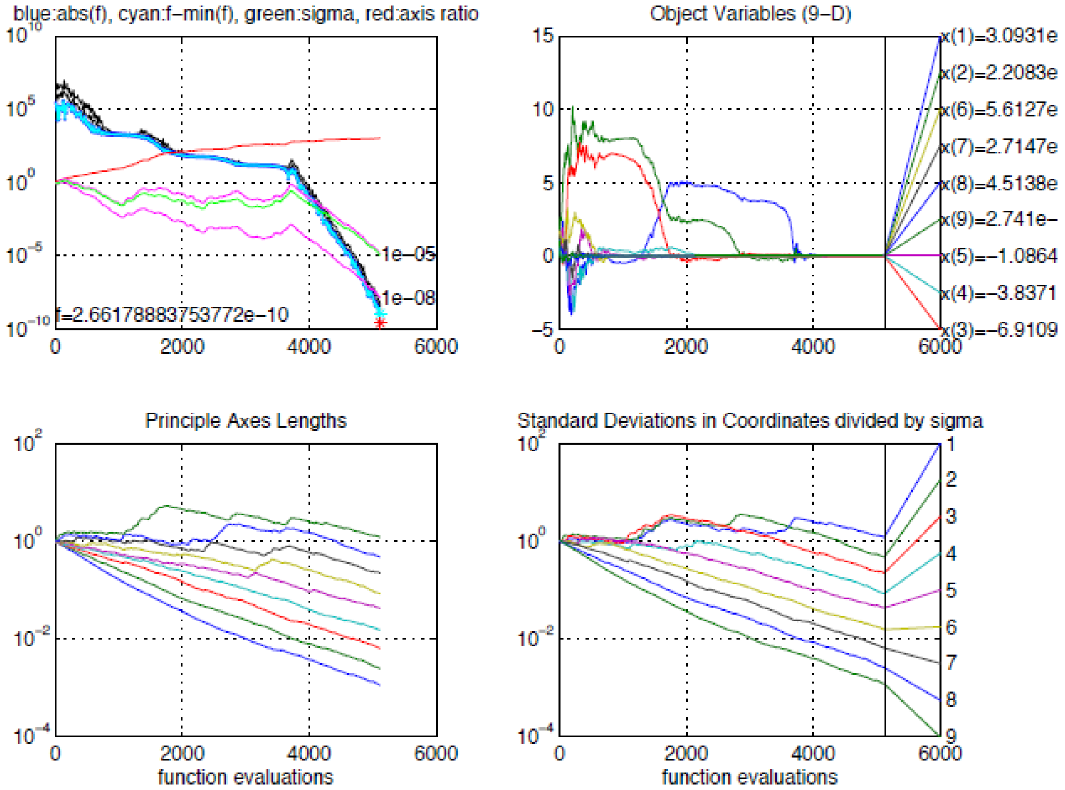
$$\mathbf{C} \propto \boldsymbol{H}^{-1}$$

in a stochastic sense

from [Hansen, p. 91]

Experimentum Crucis (1)

f convex quadratic, separable

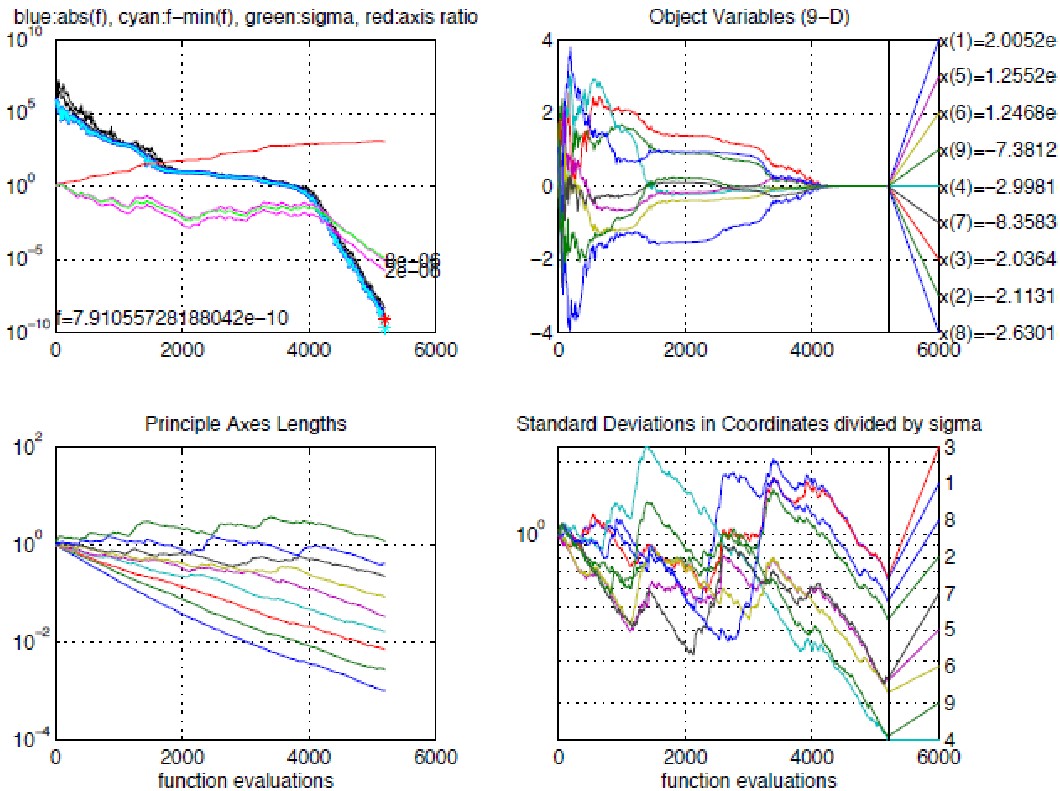


$$f(\boldsymbol{x}) = \sum_{i=1}^{n} 10^{\alpha \frac{i-1}{n-1}} x_i^2, \alpha = 6$$

from [Hansen, p. 92]

Experimentum Crucis (2)

f convex quadratic, as before but non-separable (rotated)



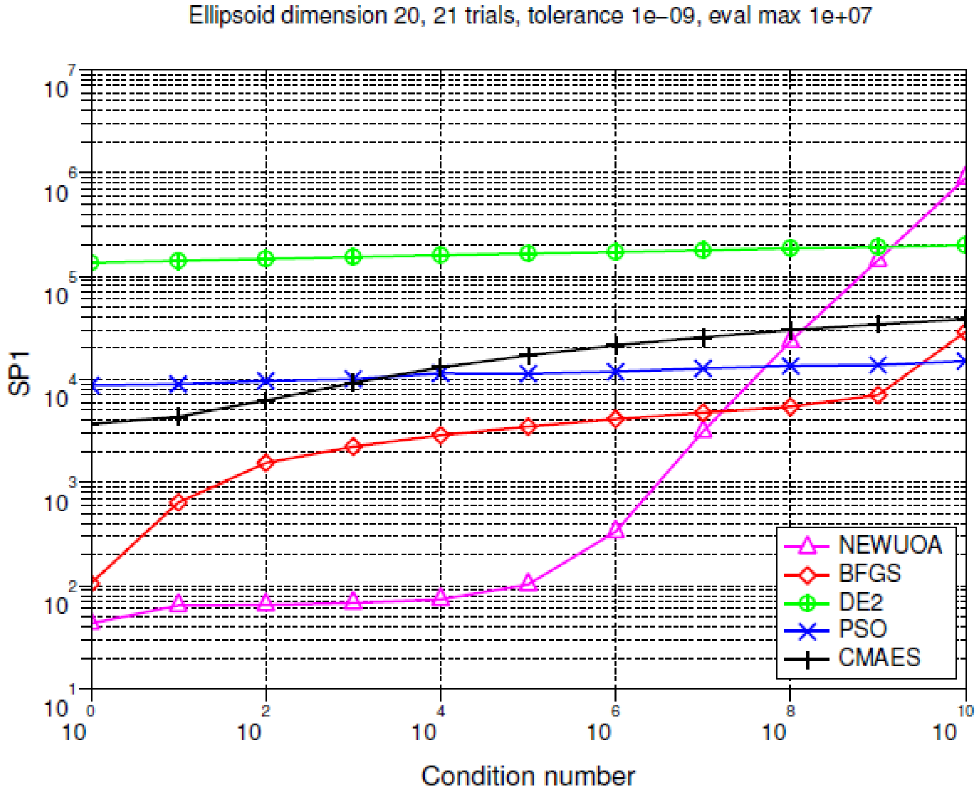
$\mathbf{C} \propto \boldsymbol{H}^{-1}$ for all $g, \mathbf{H}$

$f(\boldsymbol{x}) = g\left(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x}\right)$, $g : \mathbb{R} \to \mathbb{R}$ stricly increasing

from [Hansen, p. 93]

Comparison to BFGS, NEWUOA, PSO and DE

f convex quadratic, separable with varying condition number α



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)

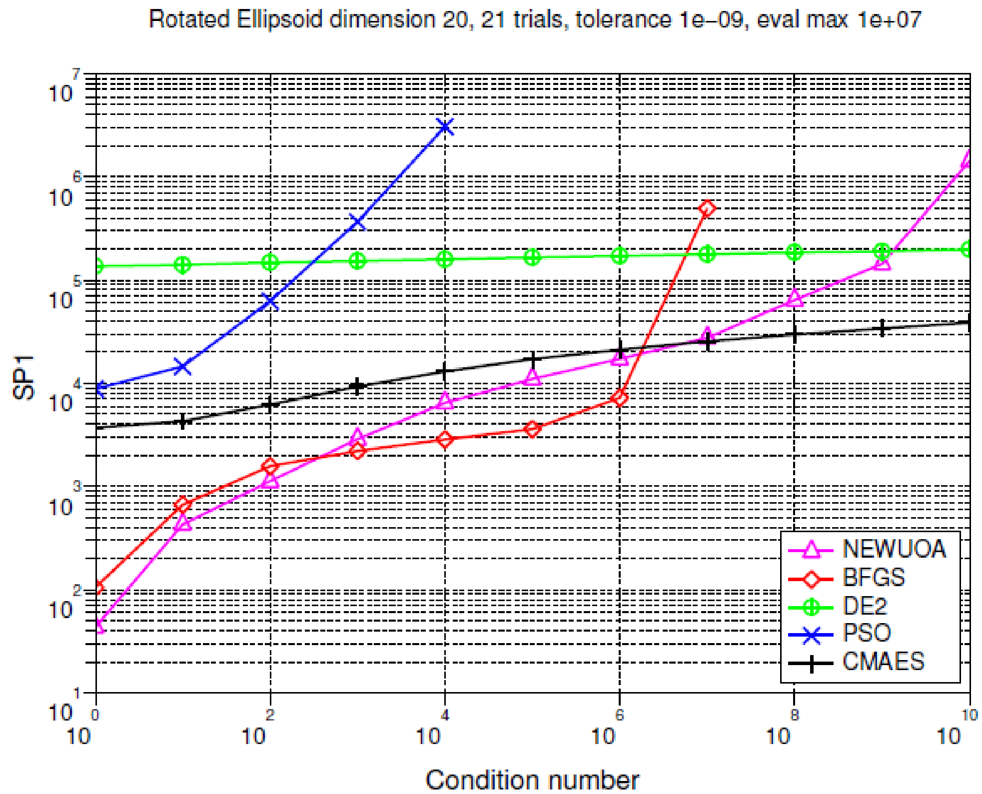
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[14] to reach the target function value of $g^{-1}(10^{-9})$

[14] Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA

Comparison to BFGS, NEWUOA, PSO and DE

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



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)

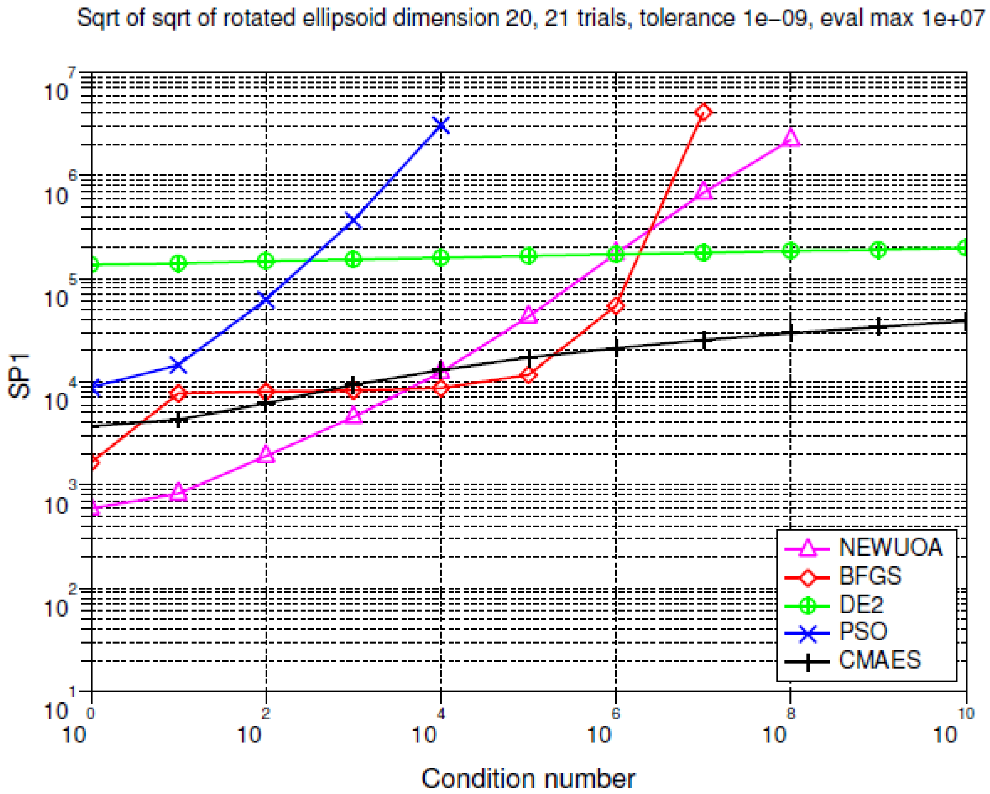
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[15] to reach the target function value of $g^{-1}(10^{-9})$

[15] Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA

Comparison to BFGS, NEWUOA, PSO and DE

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



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)

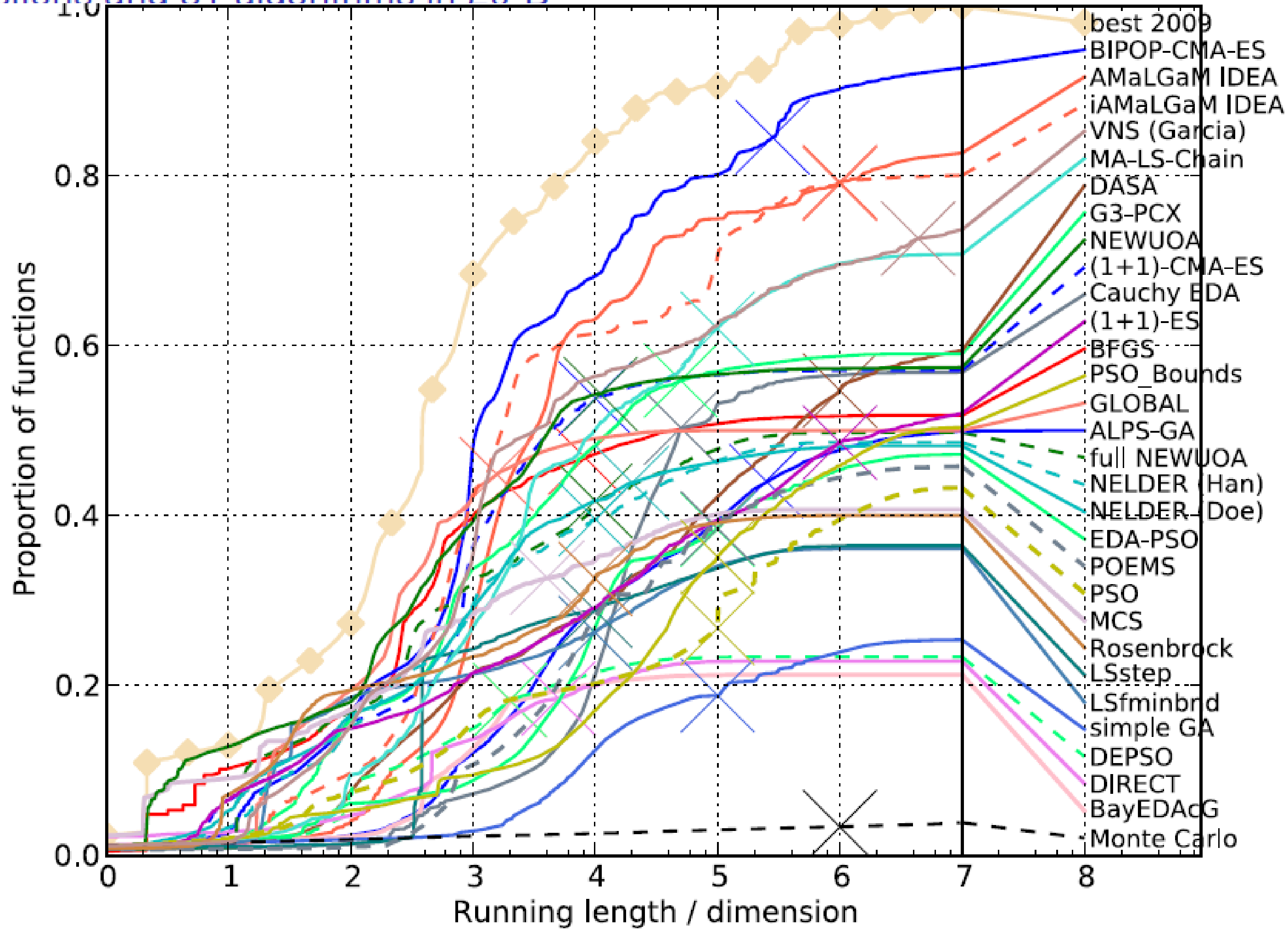
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[16] to reach the target function value of $g^{-1}(10^{-9})$

[16] Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA

Comparison during BBOB at GECCO 2009

24 functions and 31 algorithms in 20-D



Main Characteristics of (CMA) Evolution Strategies

1. Multivariate normal distribution to generate new search points
   follows the maximum entropy principle
2. Rank-based selection
   implies invariance, same performance on $g(f(\boldsymbol{x}))$ for any increasing g
   more invariance properties are featured
3. Step-size control facilitates fast (log-linear) convergence and possibly linear scaling with the dimension
   in CMA-ES based on an evolution path (a non-local trajectory)
4. *Covariance matrix adaptation (CMA)* increases the likelihood of previously successful steps and can improve performance by orders of magnitude
   the update follows the natural gradient
   $\mathbf{C} \propto \boldsymbol{H}^{-1} \iff$ adapts a variable metric
   $\iff$ new (rotated) problem representation
   $\implies f : \boldsymbol{x} \mapsto g(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x})$ reduces to $\boldsymbol{x} \mapsto \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$

Limitations

of CMA Evolution Strategies

- internal CPU-time: $10^{-8}n^2$ seconds per function evaluation on a 2GHz PC, tweaks are available
  1 000 000 f-evaluations in 100-D take 100 seconds *internal* CPU-time

- better methods are presumably available in case of
  - partly separable problems
  - specific problems, for example with cheap gradients
    specific methods
  - small dimension ($n \ll 10$)
    for example Nelder-Mead
  - small running times (number of f-evaluations $< 100n$)
    model-based methods

Conclusions

I hope it became clear...

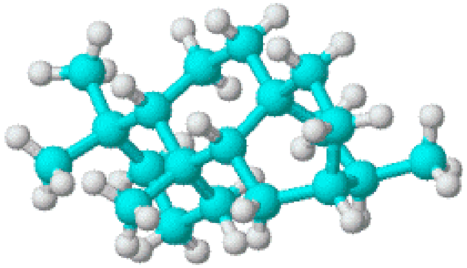
...that CMA-ES samples according to multivariate normal distributions

...how CMA-ES updates its mean, stepsize, and covariance matrix

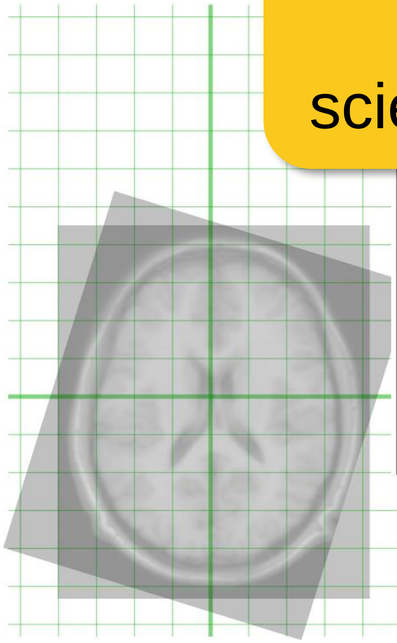
...and what are the invariance properties of CMA-ES

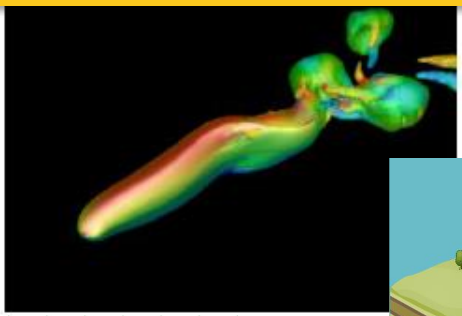
Benchmarking Optimization Algorithms

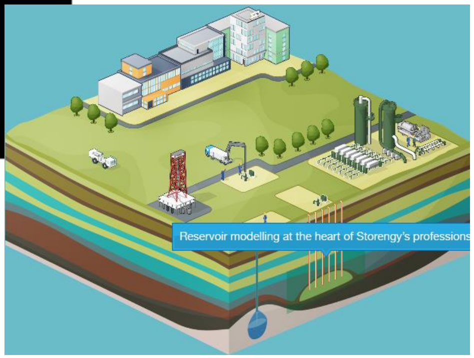
or: critical performance assessment



challenging optimization problems
appear in many
scientific, technological and industrial domains







Practical (Numerical) Blackbox Optimization

Given:

derivatives not available or not useful

Not clear:

which of the many algorithms should I use on my problem?

Numerical Blackbox Optimizers

Deterministic algorithms

Quasi-Newton with estimation of gradient (**BFGS**) [Broyden et al. 1970]
Simplex downhill [Nelder & Mead 1965]
Pattern search [Hooke and Jeeves 1961]
Trust-region methods (NEWUOA, BOBYQA) [Powell 2006, 2009]

Stochastic (randomized) search methods

Evolutionary Algorithms (continuous domain)

- Differential Evolution [Storn & Price 1997]
- Particle Swarm Optimization [Kennedy & Eberhart 1995]
- **Evolution Strategies, CMA-ES** [Rechenberg 1965, Hansen & Ostermeier 2001]
- Estimation of Distribution Algorithms (EDAs) [Larrañaga, Lozano, 2002]
- Cross Entropy Method (same as EDA) [Rubinstein, Kroese, 2004]
- Genetic Algorithms [Holland 1975, Goldberg 1989]

Simulated annealing [Kirkpatrick et al. 1983]
Simultaneous perturbation stochastic approx. (SPSA) [Spall 2000]

Need: Benchmarking

- understanding of algorithms
- algorithm selection
- putting algorithms to a standardized test
  - simplify judgement
  - simplify comparison
  - regression test under algorithm changes

Kind of everybody has to do it (and it is tedious):

- choosing (and implementing) problems, performance measures, visualization, stat. tests, ...
- running a set of algorithms

Exercise

How would you compare algorithms?

assumptions:

- continuous search space $\mathbb{R}^n$
- blackbox scenario w/o constraints
- two algorithms

a) Define a concrete experimental setup

- What to do if I want to compare algorithms A and B?
- Which experiment parameters you have to decide on?

b) What would you display to compare the performance?

c) Generalize

- arbitrary search space
- constraints
- any number of algorithms
- deterministic vs. stochastic algorithms