Tutorial: Covariance Matrix Adaptation (CMA) Evolution Strategy

Nikolaus Hansen

Institute of Computational Science
ETH Zürich

September 2006

Content

Einstein once spoke of the "unreasonable effectiveness of mathematics" in describing how the natural world works. Whether one is talking about basic physics, about the increasingly important environmental sciences, or the transmission of disease, mathematics is never any more, or any less, than a way of thinking clearly. As such, it always has been and always will be a valuable tool, but only valuable when it is part of a larger arsenal embracing analytic experiments and, above all, wide-ranging imagination.

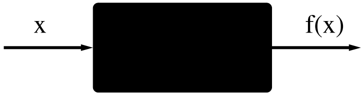
Lord Kay

Problem Statement
Continuous Domain Search/Optimization

- Continuous domain, **minimize fitness function**

$$f : \mathcal{S} \subseteq \mathbb{R}^n \to \mathbb{R}$$
$$\boldsymbol{x} \mapsto f(\boldsymbol{x})$$

- Black Box scenario



- Typical Examples:
  - shape optimization — curve fitting, airfoils
  - model calibration — biological, physical
  - parameter calibration — plants, controller, image matching

Problem Properties

We assume $f : \mathcal{S} \subset \mathbb{R}^n \to \mathbb{R}$ to have at least moderate dimensionality, say $n \not\ll 10$, and to be *non-linear, non-convex, and non-separable*.

Additionally, f can be

- multimodal
- non-smooth
- discontinuous
- ill-conditioned
- noisy
- ...

Goal: cope with these function properties that are related to real-world problems

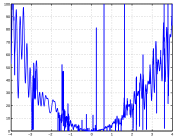
What makes a problem hard?

Why randomized search?

- ruggedness

  non-smooth, discontinuous, multimodal, and/or noisy function

- dimensionality (considerably) larger than three
- non-separability
- ill-conditioning



cut from 3-D example, solvable with standard CMA-ES

Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the **rapid increase in volume** associated with adding extra dimensions to a (mathematical) space.

Consider placing 100 points onto a real interval, say $[0, 1]$. To get similar coverage[1] of the 10-dimensional space $[0, 1]^{10}$ would require $100^{10} = 10^{20}$ points, while 100 points appear now as isolated points in a vast empty space.

Consequently, a search policy (e.g. exhaustively sample the search space volume) that is valuable in small dimensions might be useless in moderate or large dimensional search spaces.

[1] coverage in terms of distance between adjacent points

Separable Problems

Definition (Separable Problem)

A function f is separable if

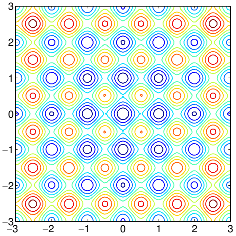
$$\left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right) = \arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n)$$

$\Rightarrow$ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example: Additively decomposable functions

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

Rastrigin function



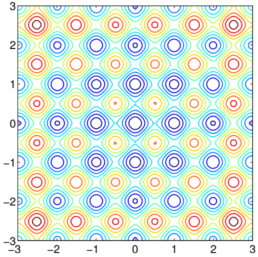
Non-Separable Problems

Building a non-separable problem from a separable one

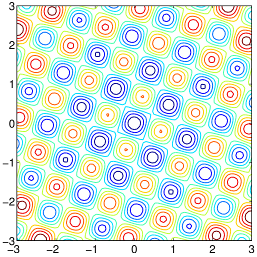
Rotating the coordinate system

- $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$ separable
- $f : \boldsymbol{x} \mapsto f(\mathbf{R}\boldsymbol{x})$ **non-separable**

$\mathbf{R}$ rotation matrix



$\xrightarrow{\mathbf{R}}$



[23]

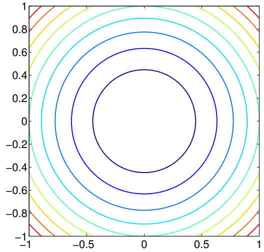
[2] Hansen, Ostermeier, Gawelczyk (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. Sixth ICGA, pp. 57-64, Morgan Kaufmann

[3] 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

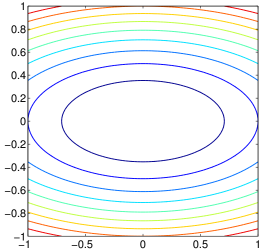
Ill-Conditioned Problems

If f is quadratic, $f : \boldsymbol{x} \mapsto \boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}$, ill-conditioned means a high condition number of Hessian Matrix $\boldsymbol{H}$

ill-conditioned means “squeezed” lines of equal function value



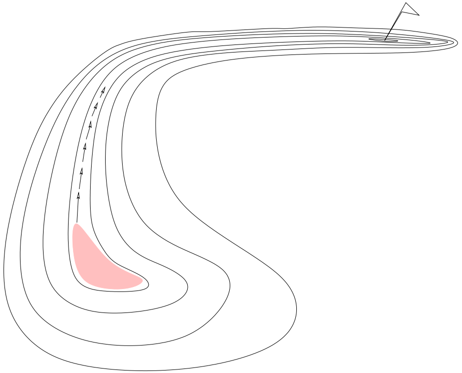
Increased $\longrightarrow$ condition number



consider the curvature of iso-fitness lines

Ill-Conditioned Problems

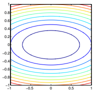
Example: A Narrow Ridge



Volume oriented search ends up in the pink area.
To approach the optimum an ill-conditioned problem needs to be solved (e.g. by following the narrow bent ridge).[4]

[4] Whitley, Lunacek, Knight 2004. Ruffled by Ridges: How Evolutionary Algorithms Can Fail, *GECCO*

Implication of Ill-Conditioning

Consider the convex quadratic function 

$$f : \mathbb{R}^n \to \mathbb{R},$$
$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^T \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}^*)$$

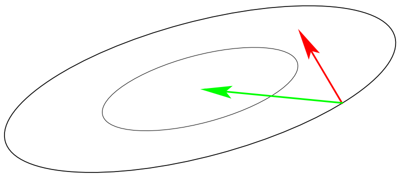
where the Hessian $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ is symmetric positive definite. We have $f \geq 0$ (by definition of positive definiteness) and the minimizer for f is $\boldsymbol{x} = \boldsymbol{x}^*$.
Differentiating the equation with respect to $\boldsymbol{x}$ yields

$$f'(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^{\mathrm{T}} \boldsymbol{H} \qquad \text{and after some rearrangement}$$
$$f'(\boldsymbol{x})^{\mathrm{T}} = \boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}\boldsymbol{x}^* \qquad \boldsymbol{H} = \boldsymbol{H}^{\mathrm{T}}, \boldsymbol{H} \text{ linear}$$
$$\boldsymbol{H}\boldsymbol{x}^* = \boldsymbol{H}\boldsymbol{x} - f'(\boldsymbol{x})^{\mathrm{T}}$$
$$\boldsymbol{x}^* = \boldsymbol{x} - \boldsymbol{H}^{-1} f'(\boldsymbol{x})^{\mathrm{T}} \qquad \boldsymbol{H} \textbf{ invertible}$$

we have solved the equation for $\boldsymbol{x}^*$ given $\boldsymbol{x}$.

The Benefit of Second Order Information

$$x^* = x - H^{-1} f'(x)^{\mathrm{T}}$$



condition number of 9

condition numbers between 100 and even 10^6 are regularly observed in real world problems

gradient direction $-f'(x)^{\mathrm{T}}$

Newton direction $-H^{-1} f'(x)^{\mathrm{T}}$

For $H \approx \mathbf{I}$ (small condition number of H) first order information (estimating the gradient) is sufficient to approach the optimum effectively. Otherwise **second order information** (estimation of H^{-1}) **is required.**

Second Order Approaches
An Overview

- quasi-Newton method
- conjugate gradients
- trust region methods
- surrogate model methods
- linkage learning
- correlated mutations (self-adaptation)
- estimation of distribution algorithms

The mutual idea

capture **dependencies** between variables, a second-order model

. . . summary

What makes a problem hard?
Summary

- ruggedness

  addressed by ES-selection and recombination

- dimensionality (considerably) larger than three
- non-separability

  addressed by covariance matrix adaptation (CMA)

- ill-conditioning

  addressed by CMA

. . . interface to real world problems

Problem Encoding

A real world problem requires

- a representation; the encoding of problem parameters into $\boldsymbol{x} \in \mathcal{S} \subset \mathbb{R}^n$
- the definition a fitness function $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$ to be minimized

One might distinguish two approaches

Natural Encoding

Use a “natural” encoding and **design the optimizer** with respect to the problem e.g. use of specific “genetic operators”

frequently done in discrete domain

Concerned Encoding

Put all problem specific knowledge into the encoding and use a **“generic” optimizer**

frequently done in continuous domain

Advantage: Sophisticated and well-validated optimizers (as CMA) can be used

Linear Encoding and the Mutation Operator

Equivalence between change in encoding and transformation of the mutation operator

Let $\boldsymbol{x}_B, \boldsymbol{x}_A \in \mathbb{R}^n$ be **two genotypes** encoding the same phenotype

$$\boldsymbol{y} = \mathbf{A}\,\boldsymbol{x}_A = \mathbf{B}\,\boldsymbol{x}_B$$

via the different linear transformations (matrices) $\mathbf{A}$ and $\mathbf{B}$. The **effect of the different encodings** becomes evident, when mutation is applied on the genotype (by adding $\mathcal{N}$).

$$\begin{array}{rcccl} \boldsymbol{y}_{\text{new}} & = & \mathbf{B}\,(\boldsymbol{x}_B + \mathcal{N}) & = & \mathbf{B}\,\boldsymbol{x}_B + \mathbf{B}\,\mathcal{N} \\ & & & = & \mathbf{A}\,\boldsymbol{x}_A + \mathbf{A}\,\mathbf{A}^{-1}\mathbf{B}\,\mathcal{N} \\ & & & = & \mathbf{A}\,(\boldsymbol{x}_A + \mathbf{A}^{-1}\mathbf{B}\,\mathcal{N}) \\ \boldsymbol{y}_{\text{new}} & = & \mathbf{A}\,(\boldsymbol{x}_A + \mathbf{A}^{-1}\mathbf{B}\,\mathcal{N}) & & \end{array}$$

Using a new encoding $\mathbf{B}$ means, in case of additive mutation, to introduce a **linear transformation $\mathbf{A}^{-1}\mathbf{B}$ for the mutation** in encoding $\mathbf{A}$. Because

$$\mathbf{A}^{-1}\mathbf{B}\,\mathcal{N}(\mathbf{0}, \mathbf{C}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{A}^{-1}\mathbf{B}\,\mathbf{C}(\mathbf{A}^{-1}\mathbf{B}\,)^{\mathrm{T}}\right)$$

this means using a different covariance matrix for the mutation operator.

The CMA Evolution Strategy

Randomized Search

A black box search template to minimize $f : \mathbb{R}^n \to \mathbb{R}$

```
Initialize distribution parameters $\boldsymbol{\theta}$, set population size $\lambda \in \mathbb{N}$
While not terminate
    1. Sample distribution $P(\boldsymbol{x}|\boldsymbol{\theta}) \to \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda \in \mathbb{R}^n$
    2. Update parameters $\boldsymbol{\theta} \leftarrow F_\theta(\boldsymbol{\theta}, \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda, f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_\lambda))$
```

Everything depends on the definition of P and F_θ (deterministic algorithms are covered as well)

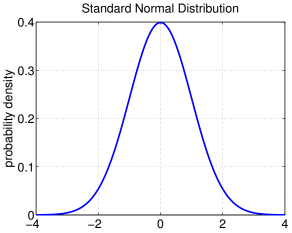
In the CMA evolution strategy

- P is a multi-variate normal distribution $\mathcal{N}$
- $\boldsymbol{\theta} = \{\boldsymbol{m}, \mathbf{C}, \sigma\}$

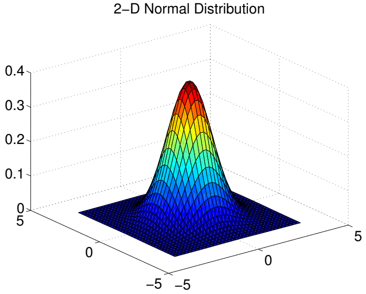
Why Normal Distributions?

1. the isotropic distribution does not (unfoundedly) favor any direction
   implies invariances
2. maximum entropy distribution with finite variance
   there are the least possible assumptions on f in the distribution shape
3. only stable distribution with finite variance
   stable means the sum of normal variates is again normal,
   helpful in design and analysis of algorithms
4. most convenient way to generate isotropic search points
5. widely observed in nature, for example with phenotypic traits

Normal Distribution



probability density of 1-D standard normal distribution



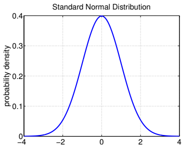
probability density of 2-D normal distribution

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 $\mathbf{C}$.

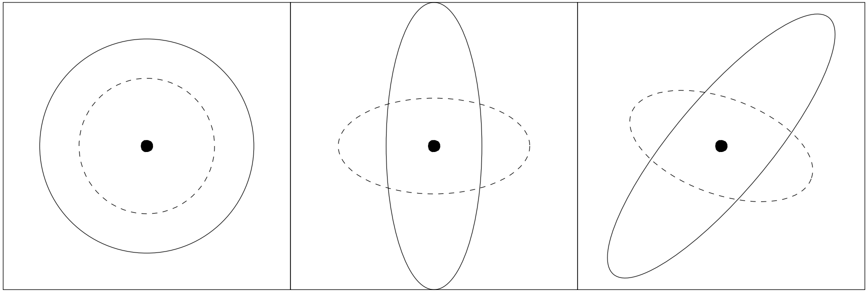
The **mean** value $\boldsymbol{m}$

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



The **covariance matrix** $\mathbf{C}$ determines the shape. It has a nice **geometrical interpretation**: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{x} = 1\}$

Lines of Equal Density



$\mathcal{N}\left(\boldsymbol{m}, \sigma^2\mathbf{I}\right) \sim \boldsymbol{m} + \sigma\mathcal{N}(\mathbf{0},\mathbf{I})$
one free parameter σ
components of $\mathcal{N}(\mathbf{0},\mathbf{I})$ are independent standard normally distributed

$\mathcal{N}\left(\boldsymbol{m}, \mathbf{D}^2\right) \sim \boldsymbol{m} + \mathbf{D}\,\mathcal{N}(\mathbf{0},\mathbf{I})$
n free parameters
components are independent, scaled

$\mathcal{N}(\boldsymbol{m}, \mathbf{C}) \sim \boldsymbol{m} + \mathbf{C}^{\frac{1}{2}}\mathcal{N}(\mathbf{0},\mathbf{I})$
$(n^2+n)/2$ free parameters
components are correlated

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

Sampling New Search Points
The Mutation Operator

New search points are sampled normally distributed

$$\boldsymbol{x}_i \sim \mathcal{N}_i\left(\boldsymbol{m}, \sigma^2\mathbf{C}\right) = \boldsymbol{m} + \sigma\,\mathcal{N}_i(\mathbf{0}, \mathbf{C}) \qquad \text{for } i = 1, \ldots, \lambda$$

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

where

- the **mean** vector $\boldsymbol{m} \in \mathbb{R}^n$ represents the favorite solution at present
- 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 question remains how to update $\boldsymbol{m}$, $\mathbf{C}$, and σ.

Update of the Distribution Mean $\boldsymbol{m}$

Selection and Recombination

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

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

$$\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda} = \boldsymbol{m} + \sigma \underbrace{\sum_{i=1}^{\mu} w_i \boldsymbol{z}_{i:\lambda}}_{=: \langle \boldsymbol{z} \rangle_{\text{sel}}}$$

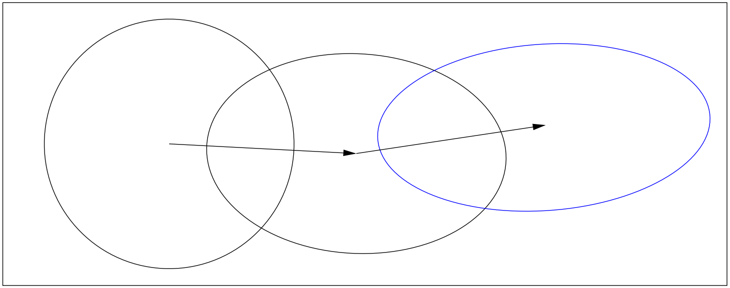
where

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

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

Covariance Matrix Adaptation
Rank-One Update

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \langle z \rangle_{\text{sel}}, \quad \langle z \rangle_{\text{sel}} = \sum_{i=1}^{\mu} w_i z_{i:\lambda}, \quad z_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$

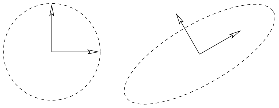


new distribution,
$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \langle z \rangle_{\text{sel}} \langle z \rangle_{\text{sel}}^{\mathrm{T}}$
the ruling principle: the adaptation increases the probability of successful steps, $\langle z \rangle_{\text{sel}}$, to appear again

The covariance matrix adaptation

- learns all pairwise dependencies between variables

  off-diagonal entries in the covariance matrix reflect the dependencies



- learns a rotated problem representation (according to the principle axes of the mutation ellipsoid)

  components are independent (only) in the new representation

- learns a scaling of the independent components

  in the new representation

- conducts a principle component analysis (PCA) of steps $\langle z \rangle_{\text{sel}}$, sequentially in time and space

  eigenvectors of the covariance matrix $\mathbf{C}$ are the principle components / the principle axes of the mutation ellipsoid

- achieves covariance matrix $\mathbf{C} \propto \boldsymbol{H}^{-1}$ on quadratic functions
- is equivalent with an adaptive (general) linear encoding[5]

. . . equations

[5] Hansen 2000, Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies, PPSN VI

Preliminary Set of Equations

Covariance Matrix Adaptation with Rank-One Update

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

$$
\begin{aligned}
\boldsymbol{x}_i &= \boldsymbol{m} + \sigma\, \boldsymbol{z}_i, \quad \boldsymbol{z}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \\
\boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_{\text{sel}} \quad \text{where } \langle \boldsymbol{z} \rangle_{\text{sel}} = \sum_{i=1}^{\mu} w_i\, \boldsymbol{z}_{i:\lambda} \\
\mathbf{C} &\leftarrow (1 - c_{\text{cov}})\mathbf{C} + c_{\text{cov}} \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \underbrace{\langle \boldsymbol{z} \rangle_{\text{sel}} \langle \boldsymbol{z} \rangle_{\text{sel}}^{\mathrm{T}}}_{\text{rank-one}}
\end{aligned}
$$

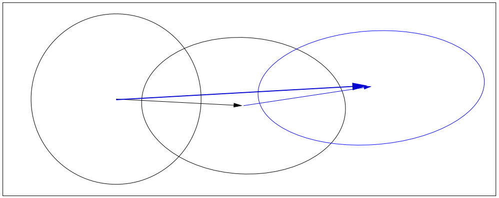
... cumulation, rank-μ, step-size control

Cumulation

The Evolution Path

Evolution Path

Conceptually, the evolution path is the 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 $\langle z \rangle_{\text{sel}}$ is used

$$\boldsymbol{p}_{\mathbf{c}} \propto \sum_{i=0}^{g} \underbrace{(1 - c_{\mathbf{c}})^{g-i}}_{\text{exponentially fading weights}} \langle z \rangle_{\text{sel}}^{(i)}$$

The recursive construction method for the evolution path is referred to as *cumulation*.

$$\boldsymbol{p}_{\mathbf{c}} \quad \leftarrow \quad \underbrace{(1 - c_{\mathbf{c}})}_{\text{decay factor}} \boldsymbol{p}_{\mathbf{c}} + \underbrace{\sqrt{1 - (1 - c_{\mathbf{c}})^2}\sqrt{\mu_{\text{eff}}}}_{\text{normalization factor}} \underbrace{\langle z \rangle_{\text{sel}}}_{\text{input}}$$

where $\mu_{\text{eff}} = \frac{1}{\sum {w_i}^2}$, $c_{\mathbf{c}} \ll 1$. History information is accumulated in the evolution path.

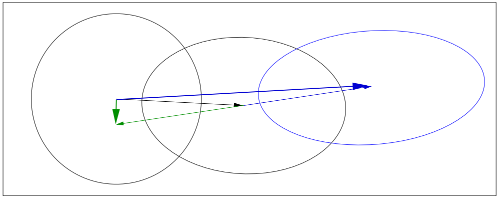
Cumulation is a common technique and also named

- exponentially weighted mooving average
- exponential smoothing (time series, forecasting)
- iterate averaging (stochastic approximation)

Cumulation

Utilizing the Evolution Path

We used $\langle z\rangle_{\mathrm{sel}}\langle z\rangle_{\mathrm{sel}}^{\mathrm{T}}$ for updating $\mathbf{C}$. Because $\langle z\rangle_{\mathrm{sel}}\langle z\rangle_{\mathrm{sel}}^{\mathrm{T}} = -\langle z\rangle_{\mathrm{sel}}(-\langle z\rangle_{\mathrm{sel}})^{\mathrm{T}}$ the sign of $\langle z\rangle_{\mathrm{sel}}$ is neglected. The sign information is (re-)introduced by using the *evolution path*.



$$p_c \quad \leftarrow \quad \underbrace{(1 - c_c)}_{\text{decay factor}}\, p_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu_{\mathrm{eff}}}}_{\text{normalization factor}}\langle z\rangle_{\mathrm{sel}}$$

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

variance effective selection mass μ_{eff}

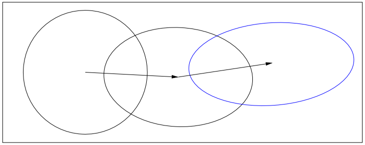
$$\begin{aligned} \boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_{\text{sel}} = \textstyle\sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda} \\ \boldsymbol{p}_{\text{c}} &\leftarrow \underbrace{(1 - c_{\text{c}})}_{\text{decay factor}} \boldsymbol{p}_{\text{c}} + \underbrace{\sqrt{1 - (1 - c_{\text{c}})^2} \sqrt{\mu_{\text{eff}}}}_{\text{normalization factor}} \langle \boldsymbol{z} \rangle_{\text{sel}} \end{aligned}$$

$$\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \qquad \text{where } \sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$$

is termed *variance effective selection mass*. It holds

$$1 \leq \mu_{\text{eff}} \leq \mu$$

it holds $\mu_{\text{eff}} = \mu$ (the number of selected points) if and only if $w_1 = w_2 = \cdots = w_\mu$.



$$\begin{aligned}
\langle z\rangle_{\text{sel}} &= \textstyle\sum_{i=1}^{\mu} w_i\, z_{i:\lambda} \quad \text{where } z_i \sim \mathcal{N}_i(\mathbf{0},\mathbf{C}) \\
\mu_{\text{eff}} &= \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \\
p_c &\leftarrow (1-c_c)\,p_c + \sqrt{1-(1-c_c)^2}\sqrt{\mu_{\text{eff}}}\,\langle z\rangle_{\text{sel}}
\end{aligned}$$

We discuss the choice of normalization constants $\sqrt{\mu_{\text{eff}}}$ and $\sqrt{1-(1-c_c)^2}$.
Under random selection the input for p_c

$$\begin{aligned}
\sqrt{\mu_{\text{eff}}}\,\langle z\rangle_{\text{sel}} &= \frac{1}{\sqrt{\sum_{i=1}^{\mu} {w_i}^2}} \sum_{i=1}^{\mu} w_i\, z_{i:\lambda} \\
&\sim \frac{1}{\sqrt{\sum_{i=1}^{\mu} {w_i}^2}} \sum_{i=1}^{\mu} w_i\, \mathcal{N}_i(\mathbf{0},\mathbf{C}) \qquad \text{random selection, } z_{i:\lambda} \sim \mathcal{N}(\mathbf{0},\mathbf{C}) \\
&\sim \frac{1}{\sqrt{\sum_{i=1}^{\mu} {w_i}^2}} \mathcal{N}\left(\mathbf{0}, \sum_{i=1}^{\mu} {w_i}^2 \mathbf{C}\right) \qquad \mathcal{N}_i() \text{ independent} \\
&\sim \mathcal{N}(\mathbf{0},\mathbf{C})
\end{aligned}$$

$$p_c \leftarrow \underbrace{(1 - c_c)}_{\text{decay factor}} p_c + \sqrt{1 - (1 - c_c)^2} \underbrace{\sqrt{\mu_{\text{eff}}}\, \langle z \rangle_{\text{sel}}}_{\sim \mathcal{N}(\mathbf{0}, \mathbf{C})}$$

The factor $\sqrt{1 - (1 - c_c)^2}$ accounts for $1 - c_c$ such that

$$(1 - c_c)^2 + \sqrt{1 - (1 - c_c)^2}^{\,2} = 1$$

Therefore $p_c \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ given previously $p_c \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ and $\sqrt{\mu_{\text{eff}}}\, \langle z \rangle_{\text{sel}} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ independently.

. . . cumulation in update of $\mathbf{C}$

Preliminary Set of Equations (2)

Covariance Matrix Adaptation, Rank-One Update with Cumulation

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0} \in \mathbb{R}^n$,
set $c_\mathbf{c} \approx 4/n$, $c_\mathrm{cov} \approx 2/n^2$
While not terminate

$$\begin{aligned}
\boldsymbol{x}_i &= \boldsymbol{m} + \sigma \boldsymbol{z}_i, \quad \boldsymbol{z}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \\
\boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_\mathrm{sel} \quad \text{where } \langle \boldsymbol{z} \rangle_\mathrm{sel} = \sum_{i=1}^{\mu} w_i \boldsymbol{z}_{i:\lambda} \\
\boldsymbol{p}_\mathbf{c} &\leftarrow (1 - c_\mathbf{c})\boldsymbol{p}_\mathbf{c} + \sqrt{1 - (1 - c_\mathbf{c})^2}\sqrt{\mu_\mathrm{eff}}\, \langle \boldsymbol{z} \rangle_\mathrm{sel} \\
\mathbf{C} &\leftarrow (1 - c_\mathrm{cov})\mathbf{C} + c_\mathrm{cov} \underbrace{\boldsymbol{p}_\mathbf{c} \boldsymbol{p}_\mathbf{c}^{\mathrm{T}}}_{\text{rank-one}}
\end{aligned}$$

$\ldots \mathcal{O}(n^2)$ to $\mathcal{O}(n)$

Using an **evolution path** for the **rank-one update** of the covariance matrix reduces the number of function evaluations to adapt to a straight ridge **from** $\mathcal{O}(n^2)$ **to** $\mathcal{O}(n)$.[a]

[a] Hansen, Müller and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation, 11(1)*, pp. 1-18

remark: the overall model complexity is n^2 but we can learn important parts of the model in time of order n

…remark on learning rates

A Remark on Learning Rates

$$p_c \quad \leftarrow \quad (1 - c_c) p_c + \ldots$$

The parameters $c_c \approx \frac{4}{n} \leq 1$ and $c_{cov} \leq 1$ are learning rates. The larger they are, the faster the learning takes place.

- for $c_c = 1$ the previous information is completely disregarded
- there can be a trade-off between *fast* and *reliable* adaptation

The reciprocal of the learning rate, e.g. $\frac{1}{c_c} \approx \frac{n}{4}$, can be interpreted as **backward time horizon**, or **life span**, or amount information stored in p_c.

- approximately 37% ($\approx \exp(-1)$) of the information in p_c is older than the backward time horizon of $n/4$ generations
- approximately 63% of the information has vanished after $n/4$ generation steps and therefore replaced by newer information

. . . rank-μ update

Rank-μ Update

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

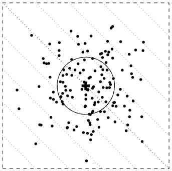
The rank-μ update extends the update rule for large population sizes using $\mu > 1$ vectors to update $\mathbf{C}$ at each generation step.
The matrix

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

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

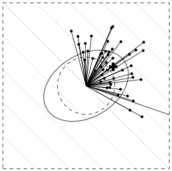
$$\mathbf{C} \leftarrow (1 - c_{\text{cov}})\, \mathbf{C} + c_{\text{cov}}\, \mathbf{Z}$$

where $c_{\text{cov}} \approx \mu_{\text{eff}} / n^2 \leq 1$.



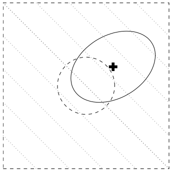
$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\, \boldsymbol{z}_i, \quad \boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$

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



$$\begin{aligned} \mathbf{Z} &= \frac{1}{\mu} \sum \boldsymbol{z}_{i:\lambda} \boldsymbol{z}_{i:\lambda}^{\mathrm{T}} \\ \mathbf{C} &\leftarrow (1-1) \times \mathbf{C} + 1 \times \mathbf{Z} \end{aligned}$$

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

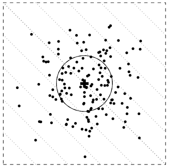


$\boldsymbol{m}_{\mathrm{new}} \leftarrow \boldsymbol{m} + \frac{1}{\mu} \sum \boldsymbol{z}_{i:\lambda}$

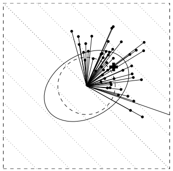
new distribution

…combined rank-one rank-μ update

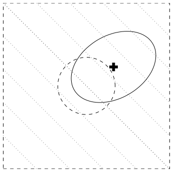
rank-μ CMA versus EMNA$_{\text{global}}$[6]



$x_i = m_{\text{old}} + z_i, \quad z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$

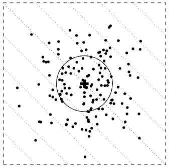


$\mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{old}})(x_{i:\lambda} - m_{\text{old}})^{\mathrm{T}}$

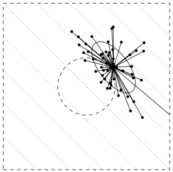


$m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum z_{i:\lambda}$

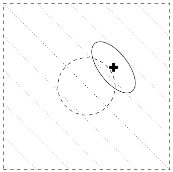
rank-μ CMA conducts a PCA **of steps**



$x_i = m_{\text{old}} + z_i, \quad z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$



$\mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{new}})(x_{i:\lambda} - m_{\text{new}})^{\mathrm{T}}$



$m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum z_{i:\lambda}$

EMNA$_{\text{global}}$ conducts a PCA **of points**

sampling of $\lambda = 150$ solutions (dots)

calculating $\mathbf{C}$ from $\mu = 50$ solutions

new distribution

The CMA-update yields a larger variance in particular in gradient direction, because m_{new} is the minimizer for the variances when calculating $\mathbf{C}$

[6] Hansen, N. (2006). The CMA Evolution Strategy: A Comparing Review. In J.A. Lozano, P. Larranga, I. Inza and E. Bengoetxea (Eds.). Towards a new evolutionary computation. Advances in estimation of distribution algorithms. pp. 75-102

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

$$\mathbf{C} = (1 - c_{\mathrm{cov}})\,\mathbf{C} + \frac{c_{\mathrm{cov}}}{\mu_{\mathrm{cov}}} \underbrace{\boldsymbol{p}_\mathrm{c}\boldsymbol{p}_\mathrm{c}^{\mathrm{T}}}_{\text{rank-one}} + c_{\mathrm{cov}} \left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right) \underbrace{\mathbf{Z}}_{\text{rank-}\mu}$$

where $\mu_{\mathrm{cov}} = \mu_{\mathrm{eff}}$.

. . . summary rank-μ

In summary

The rank-μ update

- increases the possible learning rate in large populations

  roughly from $2/n^2$ to μ_{eff}/n^2

- reduces the number of necessary **generations** roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$[7]

  given $\mu_{\text{eff}} \propto \lambda \propto n$

Therefore the rank-μ update is the primary mechanism whenever a large population size is used

say $\lambda \geq 3\,n + 10$

The rank-one update

- uses the evolution path and can learn straight ridges in $\mathcal{O}(n)$ rather than $\mathcal{O}(n^2)$ *function evaluations*.

[7] Hansen, Müller and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation, 11(1)*, pp. 1-18

Preliminary Set of Equations (3)

Rank-One Update with Cumulation, Rank-μ Update

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$,
set $c_\mathbf{c} \approx 4/n$, $c_\mathrm{cov} \approx \mu_\mathrm{eff}/n^2$, $\mu_\mathrm{cov} = \mu_\mathrm{eff}$
While not terminate

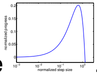
$$\begin{aligned}
\boldsymbol{x}_i &= \boldsymbol{m} + \sigma \boldsymbol{z}_i, \quad \boldsymbol{z}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), && \text{sampling} \\
\boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_\mathrm{sel} \quad \text{where } \langle \boldsymbol{z} \rangle_\mathrm{sel} = \textstyle\sum_{i=1}^{\mu} w_i \boldsymbol{z}_{i:\lambda} && \text{update mean} \\
\boldsymbol{p}_\mathbf{c} &\leftarrow (1 - c_\mathbf{c}) \boldsymbol{p}_\mathbf{c} + \sqrt{1 - (1 - c_\mathbf{c})^2} \sqrt{\mu_\mathrm{eff}} \, \langle \boldsymbol{z} \rangle_\mathrm{sel} && \text{cumulation for } \mathbf{C} \\
\mathbf{C} &\leftarrow (1 - c_\mathrm{cov}) \mathbf{C} && \text{update } \mathbf{C} \\
&\quad + c_\mathrm{cov} \frac{1}{\mu_\mathrm{cov}} \boldsymbol{p}_\mathbf{c} \boldsymbol{p}_\mathbf{c}^\mathrm{T} \\
&\quad + c_\mathrm{cov} \left(1 - \frac{1}{\mu_\mathrm{cov}}\right) \mathbf{Z} \qquad \text{where } \mathbf{Z} = \textstyle\sum_{i=1}^{\mu} w_i \boldsymbol{z}_{i:\lambda} \boldsymbol{z}_{i:\lambda}^\mathrm{T}
\end{aligned}$$

Missing: update of σ

Why Step-Size Control?

1. the covariance matrix update can hardly increase the variance in *all* directions simultaneously (that is, the *overall scale* of search, the *global step-size*, cannot be increased effectively).

   increasing the global step-size is usually required after the shape of the distribution has adapted to the function topography

2. There is a relatively small *evolution window* for the step-size 

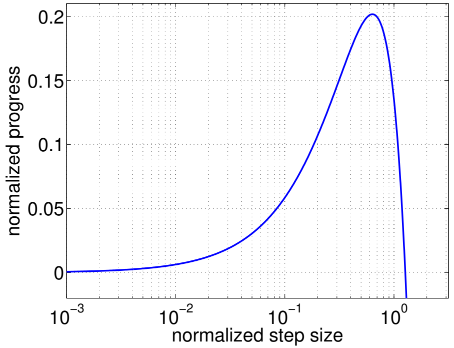
. Given $\mu \not\gg n$ the optimal step length remarkably depends on parent number μ. The classical progress theory yields

$$\sigma_{\mathrm{opt}} \propto \frac{\mu}{n}$$

   for intermediate multi-recombination, as applied in CMA, and equal recombination weights

   The length of the selected step(s) does not depend on μ in an according way. Therefore, the $\mathbf{C}$-update cannot achieve close to optimal step lengths for a wide range of μ.

Why Step-Size Control?



evolution window for the step-size on the sphere function

evolution window refers to the step-size interval where reasonable performance is observed

3. The learning rate $c_{\text{cov}} \approx \mu_{\text{eff}}/n^2$ does not comply with the requirements of convergence speed on the sphere model, $f(\boldsymbol{x}) = \sum x_i^2$. On the sphere model at least

$$\mathbf{C} \leftarrow \mathbf{C} \times \exp\left(-\frac{0.1\,\lambda}{n}\right)$$

is required to get competitive progress, given $\lambda < n$. The very best we can hope for from the $\mathbf{C}$-update is

$$\begin{aligned}\mathbf{C} \quad \leftarrow \quad & (1 - c_{\text{cov}}) \times \mathbf{C} \\ & \approx \exp(-c_{\text{cov}}) \times \mathbf{C} \\ & \approx \exp\left(-\frac{\mu_{\text{eff}}}{n^2}\right) \times \mathbf{C} \\ & \approx \exp\left(-\frac{0.3\,\lambda}{n^2}\right) \times \mathbf{C} \quad .\end{aligned}$$

Each single reason would be sufficient to ask for additional step-size control in an evolutionary algorithm

...methods for step-size control

Methods for Step-Size Control

- $1/5$-th success rule[a], applied with “+”-selection
  - increase step-size if more than 20% of the new solutions are successful, decrease otherwise
  - used in the $(1 + \lambda)$-CMA-ES[b][c]
- σ-self-adaptation[d], applied with “,”-selection
  - mutation is applied to the step-size and the better one, according to the fitness function value, is selected
- path length control[e] (cumulative step-size adaptation, CSA)[f], applied with “,”-selection
  - preferably used in the CMA evolution strategy

[a] Rechenberg 1973, *Evolutionsstrategie, Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog

[b] Igel et al 2006. A Computational Efficient Covariance Matrix Update and a (1+1)-CMA for Evolution Strategies. GECCO 2006

[c] Igel et al 2006, Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation Journal*

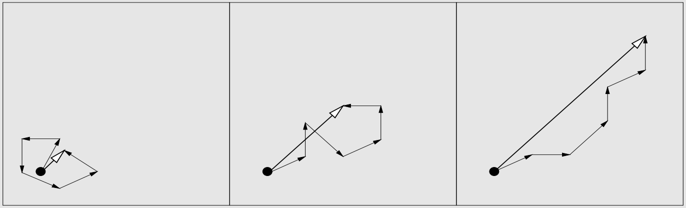
[d] Schwefel 1981, *Numerical Optimization of Computer Models*, Wiley

Path Length Control
The Concept

$$\begin{aligned} \boldsymbol{x}_i &= \boldsymbol{m} + \sigma\, \boldsymbol{z}_i \\ \boldsymbol{m} &\leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_{\text{sel}} \end{aligned}$$

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

loosely speaking steps are

- perpendicular under random selection (in expectation)
- perpendicular in the desired situation (to be most efficient)

Path Length Control
The Equations

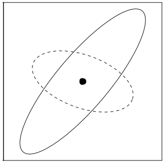
Initialize $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C}$, evolution path $\boldsymbol{p}_\sigma = \mathbf{0}$,
set $c_\sigma \approx 4/n$, $d_\sigma \approx 1$.

$$\begin{array}{rcll}
\boldsymbol{x}_i & = & \boldsymbol{m} + \sigma \boldsymbol{z}_i, \quad \boldsymbol{z}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), & \text{sampling} \\
\boldsymbol{m} & \leftarrow & \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_{\mathrm{sel}} \quad \text{where } \langle \boldsymbol{z} \rangle_{\mathrm{sel}} = \sum_{i=1}^{\mu} w_i \boldsymbol{z}_{i:\lambda} & \text{update mean} \\
\boldsymbol{p}_\sigma & \leftarrow & (1 - c_\sigma) \boldsymbol{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}}_{\text{accounts for } 1-c_\sigma} \quad \underbrace{\sqrt{\mu_{\mathrm{eff}}}}_{\text{accounts for } w_i} \quad \mathbf{C}^{-\frac{1}{2}} \langle \boldsymbol{z} \rangle_{\mathrm{sel}} & \\
\sigma & \leftarrow & \sigma \times \underbrace{\exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right) \right)}_{>1 \iff \|\boldsymbol{p}_\sigma\| \text{ is greater than its expectation}} & \text{update step-size}
\end{array}$$

where $\sqrt{\mathbf{C}^{-1}} = \mathbf{C}^{-\frac{1}{2}} = \sqrt{\mathbf{C}}^{-1}$ is symmetric positive definite

$$\begin{aligned}
\langle z\rangle_{\text{sel}} &= \textstyle\sum_{i=1}^{\mu} w_i\, z_{i:\lambda} \qquad z_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}) \\
p_\sigma &\leftarrow (1-c_\sigma)\, p_\sigma + \sqrt{1-(1-c_\sigma)^2}\sqrt{\mu_{\text{eff}}}\, \mathbf{C}^{-\frac{1}{2}} \langle z\rangle_{\text{sel}} \\
\sigma &\leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right)
\end{aligned}$$

We have $\sqrt{\mu_{\text{eff}}}\,\mathbf{C}^{-\frac{1}{2}}\langle z\rangle_{\text{sel}} = \mathbf{C}^{-\frac{1}{2}}\sqrt{\mu_{\text{eff}}}\langle z\rangle_{\text{sel}}$ and $\sqrt{\mu_{\text{eff}}}\langle z\rangle_{\text{sel}} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ under random selection.



The expected length (norm) of a $\mathcal{N}(\mathbf{0}, \mathbf{C})$ vector may strongly depend on its orientation.

The transformation $\mathbf{C}^{-\frac{1}{2}}$ aligns all directions, because

$$\mathbf{C}^{-\frac{1}{2}}\mathcal{N}(\mathbf{0}, \mathbf{C}) \sim \mathbf{C}^{-\frac{1}{2}}\mathbf{C}^{\frac{1}{2}}\mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

becomes isotropic.
Therefore, under random selection the input for p_σ

$$\sqrt{\mu_{\text{eff}}}\,\mathbf{C}^{-\frac{1}{2}}\langle z\rangle_{\text{sel}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

becomes isotropic, and $\|p_\sigma\|$ can be compared to $\mathsf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \approx \sqrt{n}$.

. . . advantages of path length control

Advantages of path length control

- on the sphere function (and therefore on any quadratic function after covariance matrix adaptation has taken place) the **target step length is close to optimal** *independent of parent number μ and of the recombination weights w_i*
- for $\mu_{\text{eff}} > 1$ and $\lambda \not\gg n$ the path length control has a considerably **larger target step length** than σ-self-adaptation[8]
- **prevents premature convergence**

  a missing control of the overall population variance (spread) is probably the most recurrent reason for premature convergence

disadvantage: the expected length of the evolution path needs to be known

[8] Hansen 1998, Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie. Eine Untersuchung zur entstochastisierten, koordinatensystemunabhängigen Adaptation der Mutationsverteilung. PhD thesis

Summary of Equations

The Covariance Matrix Adaptation Evolution Strategy

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathrm{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
set $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_\mathrm{cov} \approx \mu_\mathrm{eff}/n^2$, $\mu_\mathrm{cov} = \mu_\mathrm{eff}$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_\mathrm{eff}}{n}}$,
λ, and $w_i, i = 1, \ldots, \mu$ such that $\mu_\mathrm{eff} \approx 0.3\,\lambda$, where $\mu_\mathrm{eff} = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2}$

While not terminate

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

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{z} \rangle_\mathrm{sel} \quad \text{where } \langle \boldsymbol{z} \rangle_\mathrm{sel} = \sum_{i=1}^{\mu} w_i\, \boldsymbol{z}_{i:\lambda}$$ update mean

$$\boldsymbol{p}_\mathrm{c} \leftarrow (1 - c_\mathbf{c})\, \boldsymbol{p}_\mathrm{c} + \mathbb{1}_{\{\|\boldsymbol{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_\mathbf{c})^2} \sqrt{\mu_\mathrm{eff}}\, \langle \boldsymbol{z} \rangle_\mathrm{sel}$$ cumulation for $\mathbf{C}$

$$\mathbf{C} \leftarrow (1 - c_\mathrm{cov})\, \mathbf{C} + c_\mathrm{cov} \frac{1}{\mu_\mathrm{cov}} \boldsymbol{p}_\mathrm{c} \boldsymbol{p}_\mathrm{c}{}^\mathrm{T} + c_\mathrm{cov} \left(1 - \frac{1}{\mu_\mathrm{cov}}\right) \mathbf{Z} \quad \text{where } \mathbf{Z} = \sum_{i=1}^{\mu} w_i\, \boldsymbol{z}_{i:\lambda} \boldsymbol{z}_{i:\lambda}^\mathrm{T}$$ update $\mathbf{C}$

$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_\mathrm{eff}}\, \mathbf{C}^{-\frac{1}{2}} \langle \boldsymbol{z} \rangle_\mathrm{sel}$$ cumulation for σ

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

. . . CMA in a nutshell

Summary of Mechanisms

Covariance Matrix Adaptation ES in a Nutshell

1. Multivariate normal distribution to generate new search points
   follows the maximum entropy principle
2. Selection only based on the ranking of the f-values, weighted recombination
   using only the ranking of f-values in CMA preserves invariance
3. *Covariance matrix adaptation (CMA)* increases the probability to **repeat successful steps**
   conducts a sequential PCA
   $\Longrightarrow$ rotated problem representation
   $\Longrightarrow$ learning all pairwise dependencies
4. *Path length control* **controls the step length**
   uses the evolution path,
   aims at conjugate perpendicularity,
   non-local criterion

Adaptation of the Covariance Matrix

What do we want to achieve specifically?

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

reduce any convex quadratic function to the sphere model

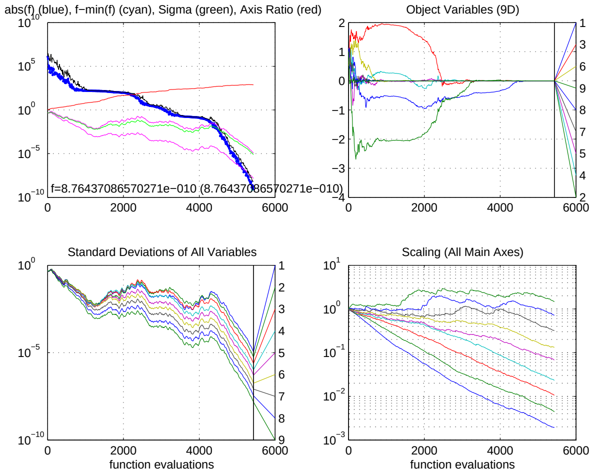
without use of derivatives

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

lines of equal density align with lines of equal fitness for quadratic f

Experimentum Crucis (1)

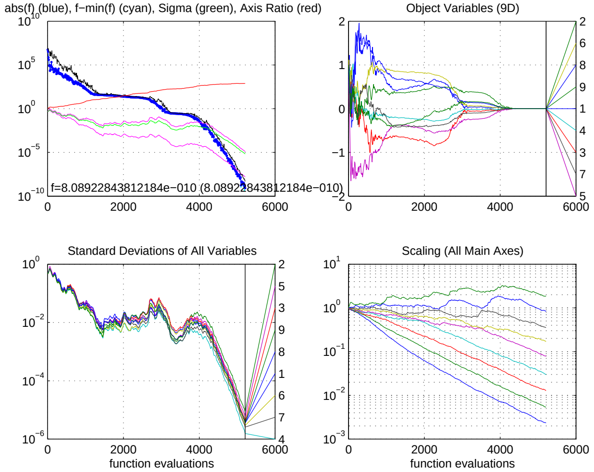
f convex quadratic, separable



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

Experimentum Crucis (2)

f convex quadratic, non-separable (rotated)



$f(\boldsymbol{x}) = g\left(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x}\right)$, $g : \mathbb{R} \to \mathbb{R}$ stricly monotonic $\Longrightarrow$ $\mathbf{C} \propto \boldsymbol{H}^{-1}$ **for all** $g, \mathbf{H}$

. . . internal parameters

Strategy 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
  - $w_{i=1,\dots,\mu}$, recombination weights

μ and w_i should be chosen such that the variance effective selection mass $\mu_{\text{eff}} \approx \frac{\lambda}{4}$, where $\mu_{\text{eff}} := 1/\sum_{i=1}^{\mu} {w_i}^2$.

- related to $\mathbf{C}$-update
  - c_{cov}, learning rate for $\mathbf{C}$-update
  - $c_{\mathbf{c}}$, learning rate for the evolution path
  - μ_{cov}, weight for rank-μ update versus rank-one update
- related to σ-update
  - c_σ, learning 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 fitness function** and are not meant to be in the users choice. Only(?) the population size λ might be reasonably varied in a wide range, *depending on the fitness function*

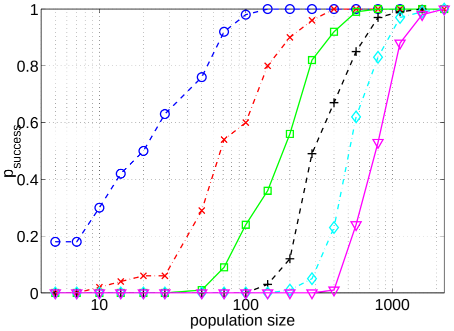
Population Size on Unimodal Functions

On unimodal functions the performance degrades at most linearly with increasing population size.

most often a small population size, $\lambda \leq 10$, is optimal

Population Size on Multi-Modal Functions

Success Probability to Find the Global Optimum



$n = 2$ ('– – ○ – –')

$n = 5$ ('– · – × – · –')

$n = 10$ ('—□—')

$n = 20$ ('– – + – –')

$n = 40$ ('– · –◇ – ·–')

$n = 80$ ('—▽—')

Shown: **success rate** versus offspring population size on the highly multi-modal Rastrigins function[9]

On multi-modal functions increasing the population size can sharply increase the success probability to find the global optimum

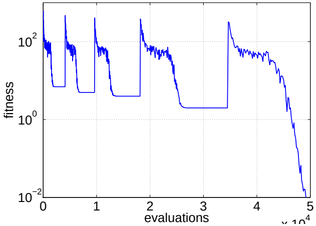
[9] Hansen & Kern 2004. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. PPSN VIII, Springer-Verlag, pp. 282-291.

Multistart With Increasing Population Size

Increase by a Factor of Two Each Restart

1. no performance loss, where small population size is sufficient (e.g. on unimodal functions)
2. moderate performance loss, if large population size is necessary

loss has an upper bound



for a factor between successive runs of ≥ 1.5 we have a performance loss smaller than $\sum_{k=0}^{\infty} 1/1.5^k = 3$

This results in a **quasi parameter free search algorithm.**

. . . design principles

Key Points and Design Principles
Key Points

- Adaptation of a step-size σ, and different adaptation principles for $\mathbf{C}$ and σ.
  - the update of $\boldsymbol{m}$ and $\mathbf{C}$ follows the maximum likelihood principle

$$\text{choose } \boldsymbol{m} \text{ such that } Prob\left(\boldsymbol{x}_{\text{sel}} \middle| \mathcal{N}\left(\boldsymbol{m}, \sigma^2\mathbf{C}\right)\right) \longrightarrow \text{ max}$$

$$\text{update } \mathbf{C} \text{ such that } Prob\left(\frac{\boldsymbol{x}_{\text{sel}} - \boldsymbol{m}_{\text{old}}}{\sigma} \middle| \mathcal{N}(\mathbf{0}, \mathbf{C})\right) \longrightarrow \text{ max}$$

under consideration of the prior $\mathbf{C}$

  - the update of σ aims at conjugate perpendicularity of consecutive steps $\Delta\boldsymbol{m}$

prevents premature convergence

Key Points

- separate learning rates for $\boldsymbol{m}$, $\mathbf{C}$, and σ
  - $\boldsymbol{m}$ is set to the best recent points. No prior information is used.
    the learning rate is determined by λ and μ_{eff}
  - $\mathbf{C}$ is controlled by the learning rate $c_{\text{cov}} \approx \mu_{\text{eff}}/n^2$. Usually much prior information is preserved, some is depleted.
  - σ is changed on a time scale of $\propto n$ generations to achieve competitive performance on the sphere model.

  yield reliable parameter adaptation to complex topologies with a small population while the performance on simple functions (where the adaptation is superfluous) is not effected
- the population size can be set without reference to learning reliability for the distribution.
  - in contrast, the *learning reliability* of most EAs, that learn a distribution, decisively depends on population size (and fitness function)

Design Principles

- stationarity for $\boldsymbol{m}$, $\mathbf{C}$, and σ under random selection. The parameters are not subject to a systematic drift, means they are unbiased in that
  - $\mathsf{E}(\boldsymbol{m}|\boldsymbol{m}_{\text{old}}) = \boldsymbol{m}_{\text{old}}$
  - $\mathsf{E}(\mathbf{C}|\mathbf{C}_{\text{old}}) = \mathbf{C}_{\text{old}}$
  - $\mathsf{E}(\log\sigma|\sigma_{\text{old}}) = \log\sigma_{\text{old}}$

$\boldsymbol{m} \leftarrow \boldsymbol{m}_{\text{old}} + \sigma\langle\boldsymbol{z}\rangle_{\text{sel}}$

this implies $\mathsf{E}(\log\sigma^\alpha|\sigma_{\text{old}}) = \log\sigma^\alpha_{\text{old}}$ for all $\alpha > 0$
and $\mathsf{E}(\sigma^\alpha|\sigma_{\text{old}}) \geq \sigma^\alpha_{\text{old}}$

- the least possible parameter tuning requirement, given a specific problem to solve

the (initial) search space region needs to be specified (initial setting of $\boldsymbol{m}$ and σ)
eventually stopping criteria need to be accommodated

- achieve or preserve as many invariances as possible

Invariance
Motivation

- empirical performance results, for example
  - from benchmark functions,
  - from solved real world problems,

  are only useful if they do generalize to other problems

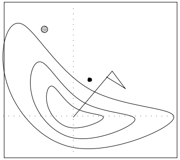
- **Invariance** is a strong **non-empirical** statement about the feasibility of generalization

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

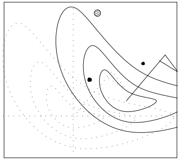
Basic Invariance in Search Space

- translation invariance

is true for most optimization algorithms



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



Identical behavior on f and $f_{\boldsymbol{a}}$

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

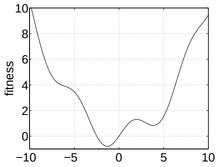
No difference can be observed w.r.t. the argument of f

Only useful if the initial point is not decisive

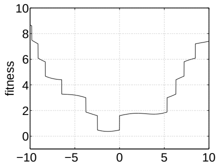
Invariance in Function Space

- invariance to order preserving transformations

preserved by ranking based selection



$$f(\boldsymbol{x}) \leftrightarrow g(f(\boldsymbol{x}))$$



Identical behavior on f and $g \circ f$ for all order preserving $g : \mathbb{R} \longrightarrow \mathbb{R}$ (strictly monotonically increasing g)

$$\begin{aligned} f : \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0 \\ g \circ f : \quad & \boldsymbol{x} \mapsto g(f(\boldsymbol{x})), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0 \end{aligned}$$

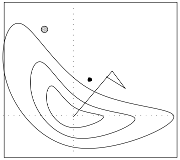
No difference can be observed w.r.t. the argument of f

Rotational Invariance in Search Space

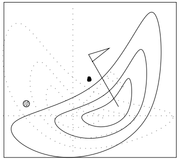
- invariance to an orthogonal transformation $\mathbf{R}$, where $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$

  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 $f_{\mathbf{R}}$

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

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

Invariances in Search Space

- invariance to any rigid (scalar product preserving) transformation in search space $\boldsymbol{x} \mapsto \mathbf{R}\boldsymbol{x} - \boldsymbol{a}$, where $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$

  e.g. true for simple evolution strategies

- scale invariance (scalar multiplication)

  exploited by step-size control

- invariance to a general linear transformation $\mathbf{G}$

  exploited by CMA

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

$$
\begin{array}{lll}
f: \quad \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} = \boldsymbol{x}_0, & \mathbf{C}^{(t=0)} = \mathbf{I} \\
f_{\mathbf{G}}: \quad \boldsymbol{x} \mapsto f(\mathbf{G}(\boldsymbol{x} - \boldsymbol{b})), & \boldsymbol{x}^{(t=0)} = \mathbf{G}^{-1}\boldsymbol{x}_0 + \boldsymbol{b}, & \mathbf{C}^{(t=0)} = \mathbf{G}^{-1}{\mathbf{G}^{-1}}^{\mathrm{T}}
\end{array}
$$

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

Only useful with an effective adaptation of $\mathbf{C}$

Invariance
Summary

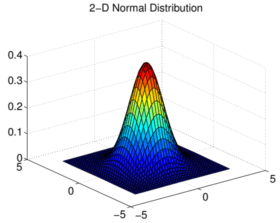
- Invariance introduces **equivalence classes** of functions, where **identical performance** must be observed on a complete function class
- **Invariance** is a strong **non-empirical** statement about the feasibility of generalization

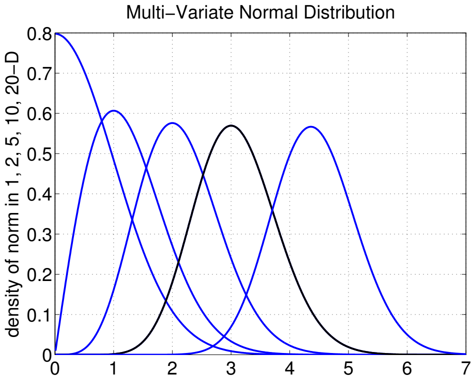
  rotation has $\frac{n^2-n}{2}$ free parameters
- The CMA Evolution Strategy **inherits all invariances** from simple evolution strategies (to *rigid transformations* of the search space and to *order preserving transformations* of the function value)
- The CMA adds invariance to general linear transformations

  only useful *together* with the effective adaptation of the covariance matrix

Normal Distribution Revisited

While the maximum likelihood of the multi-variate normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$ is at zero, the distribution of its norm $\|\mathcal{N}(\mathbf{0},\mathbf{I})\|$ reveals a different, surprising picture.





- In 10-D (black) the usual step length is about $3 \times \sigma$ and step lengths smaller than $1 \times \sigma$ virtually never occur
- Remind: this norm-density shape maximizes the distribution entropy

Performance Evaluation

Evaluation of the performance of a search algorithm needs

- meaningful quantitative measure on benchmark functions or real world problems
- acknowlegde invariance properties
- account for meta-parameter tuning
- account for algorithm internal cost, depending on the fitness function cost
  On a 2.5GHz processor our CMA-ES implementation needs
  - roughly $3 \times 10^{-8}(n+4)^2$ seconds per function evaluation
  - for one million function evaluations roughly

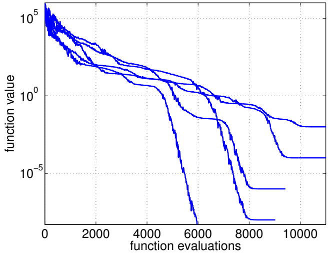
n	time
10	5s
30	30s
100	300s

Performance Measure

We have to record

- goodness of the solution
  the fitness function value
- cost of search
  number of function evaluations

fix one and measure the other

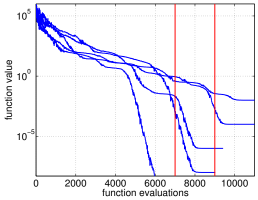


Performance Measure

fixed number of function evaluations

vertical view

measure: statistics of fitness function values



advantages

- fixing the number of function evaluations is easy

disadvantages

- assessment of **short runs**, i.e. runs that reach the global optimum *before* the given number of function evaluations
- the result is usually **not a quantitative measure**

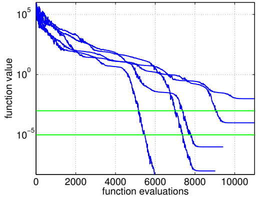
ordinal (rank) scale is available, ratio scale (“A is twice as good as B”) would be desired

Performance Measure

fixed function value

horizontal view

measure: statistics of the number of function evaluations



advantages

- comprehensible and well interpretable measure

measuring the cost

- **quantitative measure** (on the ratio scale)
- algorithm-internal CPU costs can be easily included

disadvantages

- assessment of “long” runs that do not obtain the given function value

A Comparison Study With 11 Evolutionary Algorithms
Methods

- task: black-box optimization of 25 benchmark functions
- 25 runs on each benchmark function for each dimension $n = 10, 30$
- a run is **successful** if the global optimum is reached with the given precision, before the
- maximum number of function evaluations
$$\mathtt{FE}_{\text{max}} = \begin{cases} 10^5 & \text{for } n = 10 \\ 3 \times 10^5 & \text{for } n = 30 \end{cases} \text{ is reached}$$

Remark: the setting of $\mathtt{FE}_{\text{max}}$ has a remarkable influence on the results, if the target function value can be reached only for a (slightly) larger number of function evaluations with a high probability.
Where $\mathtt{FEs} \geq \mathtt{FE}_{\text{max}}$ the result must be taken with great care.

Reference

Suganthan, Hansen, Liang, Deb, Chen, Auger, and Tiwari (2005). *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*, Technical report, Nanyang Technological University, Singapore, May 2005, http://www.ntu.edu.sg/home/EPNSugan

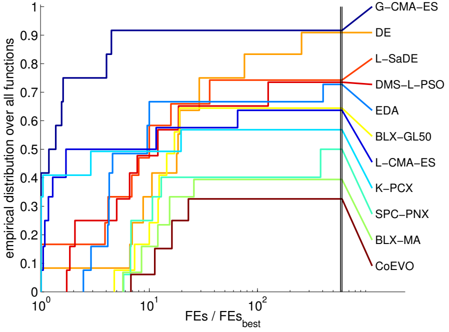
References to Algorithms

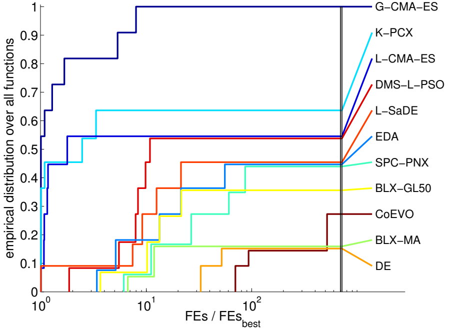
BLX-GL50	García-Martínez and Lozano (Hybrid Real-Coded. . .)
BLX-MA	Molina et al. (Adaptive Local Search. . .)
CoEVO	Pošík (Real-Parameter Optimization. . .)
DE	Rönkkönen et al. (Real-Parameter Optimization. . .)
DMS-L-PSO	Liang and Suganthan (Dynamic Multi-Swarm. . .)
EDA	Yuan and Gallagher (Experimental Results. . .)
G-CMA-ES	Auger and Hansen (A Restart CMA. . .)
K-PCX	Sinha et al. (A Population-Based,. . .)
L-CMA-ES	Auger and Hansen (Performance Evaluation. . .)
L-SaDE	Qin and Suganthan (Self-Adaptive Differential. . .)
SPC-PNX	Ballester et al. (Real-Parameter Optimization. . .)

In: CEC 2005 IEEE Congress on Evolutionary Computation, Proceedings

Summarized Results
Empirical Distribution of Normalized Success Performance

$n = 10$ $n = 30$





$\texttt{FEs} = \texttt{mean}(\#fevals) \times \frac{\text{\#all runs (25)}}{\text{\#successful runs}}$, where $\#fevals$ includes only successful runs.

Shown: **empirical distribution function** of the Success Performance FEs divided by FEs of the best algorithm on the respective function.

Results of all functions are used where at least one algorithm was successful at least once, i.e. where the target function value was reached in at least one experiment (out of 11×25 experiments).

Small values for FEs and therefore large (cumulative frequency) values in the graphs are preferable.

Function Sets

We split the function set into three subsets

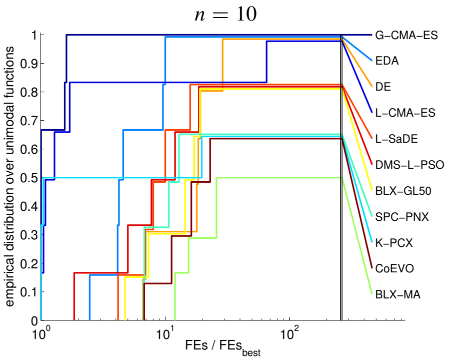
- unimodal functions
- solved multimodal functions

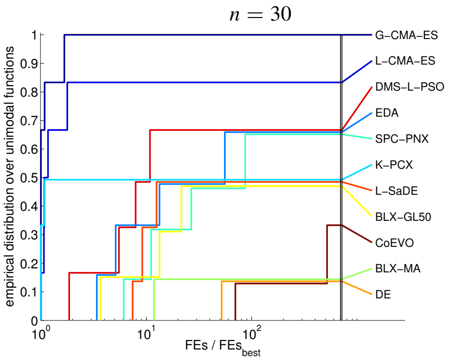
  at least one algorithm conducted at least one successful run
- unsolved multimodal functions

  no single run was successful for any algorithm

Unimodal Functions

Empirical Distribution of Normalized Success Performance





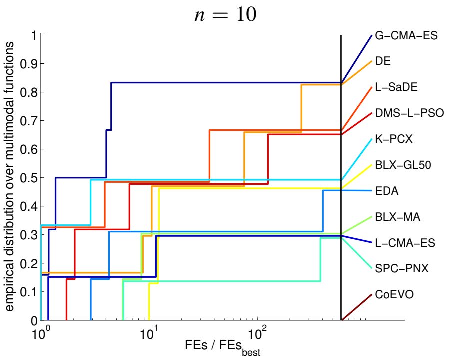
Empirical distribution function of the Success Performance FEs divided by FEs of the best algorithm (table entries of last slides).

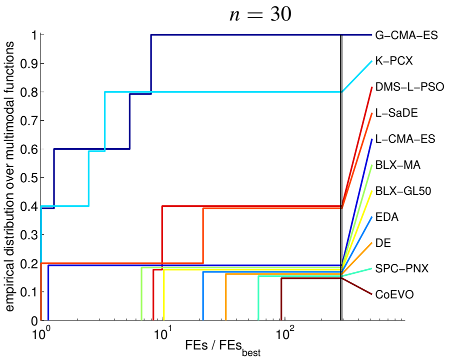
$\texttt{FEs} = \texttt{mean}(\#\mathit{fevals}) \times \frac{\#\text{all runs (25)}}{\#\text{successful runs}}$, where $\#\mathit{fevals}$ includes only successful runs.

Small values of FEs and therefore large values in the empirical distribution graphs are preferable.

Multimodal Functions

Empirical Distribution of Normalized Success Performance





Empirical distribution function of the Success Performance FEs divided by FEs of the best algorithm (table entries of last slides).

$\texttt{FEs} = \texttt{mean}(\#\mathit{fevals}) \times \frac{\#\text{all runs (25)}}{\#\text{successful runs}}$, where $\#\mathit{fevals}$ includes only successful runs.

Small values of FEs and therefore large values in the empirical distribution graphs are preferable.

Comparison Study
Conclusion

- The G-CMA-ES seems superior from three perspectives
  - performed best over all functions and on the function subsets
    - unimodal functions
    - solved multimodal functions
    - unsolved multimodal functions
  - no parameter tuning were conducted
  - most invariance properties together with EDA and K-PCX
- on two separable problems it is considerably outperformed
- The CMA-ES contradicts two myths
  - Myth 1: adaptation (of the covariance matrix) to the (local) function topography jeopardizes global search properties
  - Myth 2: a single peak multi-variate Gaussian distribution must be inferior in solving multi-modal (global) optimization problems

the curse of dimensionality limits the effectiveness of multimodal search distributions, and clustering or niching approaches

Overall Conclusion

The Take Home Message

- Important features of the CMA evolution strategy
  - learns second order information efficiently *and* reliably with small *and* large population size
  - invariance to order preserving transformations of f and invariance to rigid search space transformations
  - quasi parameter free
- CMA-ES is a robust local search algorithm
  - BFGS is roughly ten times faster on convex quadratic f
  - CMA is much more robust in a non-convex or rugged search landscape
- CMA-ES is a robust global search algorithm

  empirically outperformes most EAs on most functions
- More than 50 (successful) real-world applications

  easily applicable and often successful
- Ongoing research
  - multiobjective CMA
  - CMA in uncertain environments

References

- Main publications about CMA:

Igel, C., N. Hansen, and S. Roth (2007). *Covariance Matrix Adaptation for Multi-objective Optimization.* Evolutionary Computation, accepted for publication.

Auger, A, and Hansen, N. (2005). *A Restart CMA Evolution Strategy With Increasing Population Size.* In Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp.1769-1776.

Hansen, N, S. Kern (2004). *Evaluating the CMA Evolution Strategy on Multimodal Test Functions.* In Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings, pp. 282-291, Berlin: Springer.

Hansen, N., S.D. Müller and P. Koumoutsakos (2003) , *Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES).* Evolutionary Computation, 11(1), pp. 1-18.

Hansen, N. and A. Ostermeier (2001). *Completely Derandomized Self-Adaptation in Evolution Strategies.* Evolutionary Computation, 9(2), pp. 159-195.

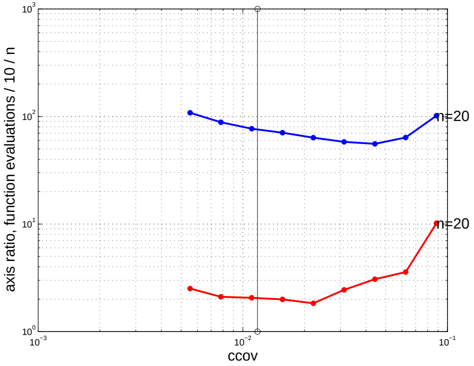
- Tutorial
- Source code in Matlab, Octave and Java
- For all this ... see

`http://www.bionik.tu-berlin.de/user/niko/`

Thank You

Determining Learning Rates

Learning rate for the covariance matrix



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

optimal condition number for $\mathbf{C}$ is one,

initial condition number of $\mathbf{C}$ equals 10^4

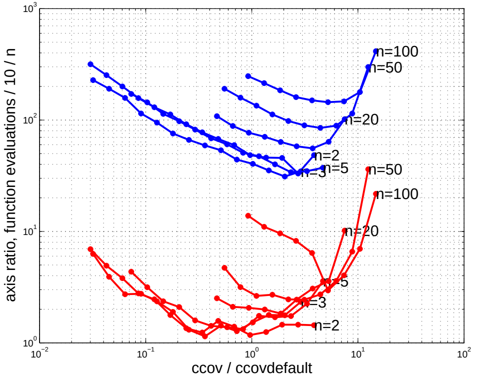
shown are single runs

x-axis: learning rate for the covariance matrix

y-axis: square root of final condition number of $\mathbf{C}$ (red), number of function evaluations to reach f_{stop} (blue)

Determining Learning Rates

Learning rate for the covariance matrix



- learning rates can be identified on simple functions

  exploiting invariance properties
- the outcome depends on the problem dimensionality
- the specific fitness function is rather insignificant

x-axis: factor for learning rate for the covariance matrix

y-axis: square root of final condition number of $\mathbf{C}$ (red), number of function evaluations to reach f_{stop} (blue)

. . . step size control

EMNA versus CMA

Both algorithms use the same sample distribution

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

In EMNA$_{\text{global}}$ $\sigma \equiv 1$ and

$$\boldsymbol{m} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \boldsymbol{x}_{i:\lambda}$$

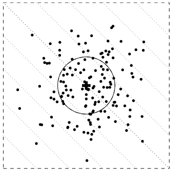
$$\mathbf{C} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} (\boldsymbol{x}_{i:\lambda} - \boldsymbol{m})(\boldsymbol{x}_{i:\lambda} - \boldsymbol{m})^{\mathrm{T}}$$

In CMA, for $c_{\text{cov}} = 1$, with rank-μ update only

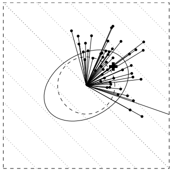
$$\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i \, \boldsymbol{x}_{i:\lambda}$$

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

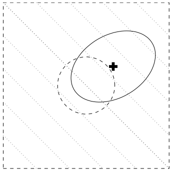
where $\boldsymbol{z}_{i:\lambda} = \frac{\boldsymbol{x}_{i:\lambda} - \boldsymbol{m}_{\text{old}}}{\sigma}$



$x_i = m_{\text{old}} + z_i, \quad z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$

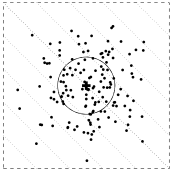


$\mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{old}})(x_{i:\lambda} - m_{\text{old}})^{\text{T}}$

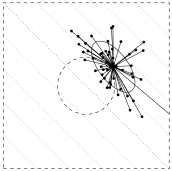


$m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum z_{i:\lambda}$

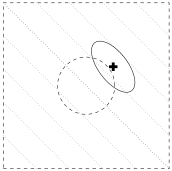
rank-μ CMA conducts a PCA **of steps**



$x_i = m_{\text{old}} + z_i, \quad z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$



$\mathbf{C} \leftarrow \frac{1}{\mu} \sum (x_{i:\lambda} - m_{\text{new}})(x_{i:\lambda} - m_{\text{new}})^{\text{T}}$



$m_{\text{new}} = m_{\text{old}} + \frac{1}{\mu} \sum z_{i:\lambda}$

EMNA$_{\text{global}}$ conducts a PCA **of points**

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

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

new distribution

the CMA-update yields a larger variance in particular in gradient direction

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