# The difficulties of black-box optimization and a stochastic variable metric approach

Nikolaus Hansen

INRIA – Saclay, TAO research group

November 9, 2009

slides: http://www.lri.fr/˜hansen/ParisRoc-handout.pdf

# 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

- Task: **minimize** an **objective function** (*fitness* function, *loss* function, *cost* function) in continuous domain

$$f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}, \qquad \underline{x} \mapsto f(\underline{x})$$

- **Black Box** scenario (direct search scenario)


  - gradients are not available or not useful
  - problem domain specific knowledge is used only within the black box, e.g. within an appropriate encoding
- Search costs: number of function evaluations

# Problem Statement

## Continuous Domain Search/Optimization

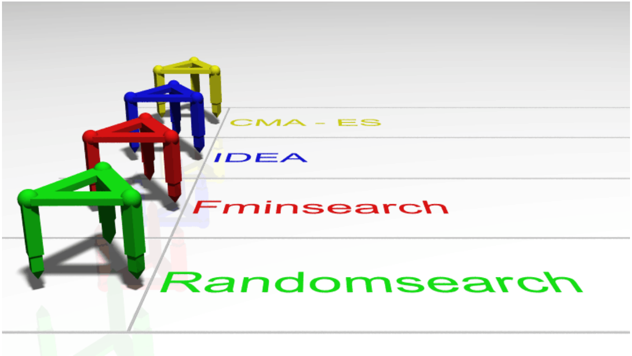
- Goal
  - solution $\underline{x}$ with **small function value** with **least search cost**
    there are two conflicting objectives
  - fast convergence to the global optimum
    ...or to a robust solution $\underline{x}$

- Typical Examples
  - shape optimization (e.g. using CFD) — curve fitting, airfoils
  - model calibration — biological, physical
  - parameter calibration — algorithms, controllers, plants, images

**Approach**: stochastic search, Evolutionary Algorithms

...tripeds

# Comparison of CMA-ES, IDEA and Simplex-Downhill




CMA-ES: Covariance Matrix Adaptation Evolution Strategy
IDEA: Iterated Density-Estimation Evolutionary Algorithm[1]
Fminsearch: Nelder-Mead simplex downhill method[2]

P. Dürr and A. Pfister (2004), Optimization of Walking Gaits for a Three Legged Robot, term paper.

see... `http://www.icos.ethz.ch/cse/research/highlights/research_highlights_august_2004`

1 Bosman (2003) Design and Application of Iterated Density-Estimation Evolutionary Algorithms. PhD thesis.
2 Nelder and Mead (1965). A simplex method for function minimization. *Computer Journal*.

...metaphores

# What Makes a Function Difficult to Solve?

Why stochastic search?

- non-linear, non-quadratic, non-convex

  on linear/quadratic functions better search policies are available

- dimensionality

  (considerably) larger than three

- non-separability

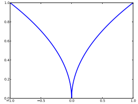
  dependencies between the objective variables

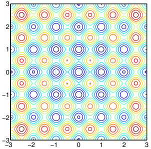
- ill-conditioning

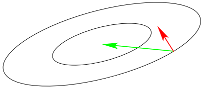
  widely varying sensitivity

- ruggedness

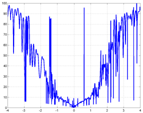
  non-smooth, discontinuous, multimodal, and/or noisy function





gradient direction Newton direction



# Curse of Dimensionality

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

**Example**: Consider placing 100 points onto a real interval, say $[0, 1]$. To get **similar coverage**, in terms of distance between adjacent points, of the $10$-dimensional space $[0, 1]^{10}$ would require $100^{10} = 10^{20}$ points. A 100 points have minimal distance of $\approx 0.65$ (on average) and appear now as isolated points in a vast empty space.

**Implication**: A **search policy** (e.g. exhaustive search) that is **efficient** in small dimensions **might be useless** in moderate or large dimensional search spaces.

# What Makes a Function Difficult to Solve?

Why stochastic search?

- non-linear, non-quadratic, non-convex

  on linear/quadratic functions better search policies are available

- dimensionality

  (considerably) larger than three

- non-separability

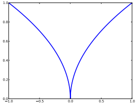
  dependencies between the objective variables

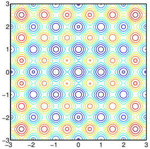
- ill-conditioning

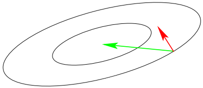
  widely varying sensitivity

- ruggedness

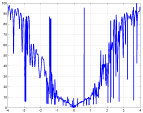
  non-smooth, discontinuous, multimodal, and/or noisy function





gradient direction Newton direction



# Separable Problems

## Definition (Separable Problem)

A function $f$ is separable if

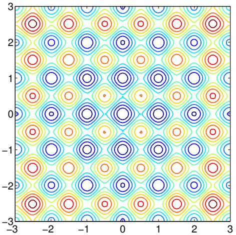
$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n) = \left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right)$$

$\Rightarrow$ 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



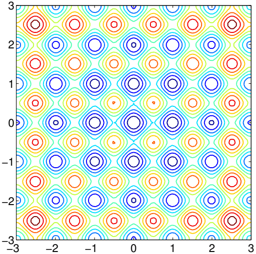
# Non-Separable Problems

## Building a non-separable problem from a separable one

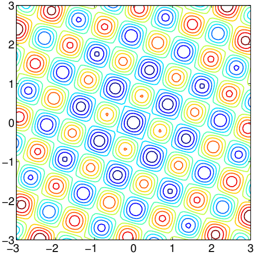
### Rotating the coordinate system

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

$\underline{\underline{R}}$ rotation matrix




$$\xrightarrow{\underline{\underline{R}}}$$




[3,4]

[3] 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

[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

# What Makes a Function Difficult to Solve?

Why stochastic search?

- non-linear, non-quadratic, non-convex

  on linear/quadratic functions better search policies are available

- dimensionality

  (considerably) larger than three

- non-separability

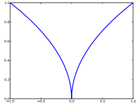
  dependencies between the objective variables

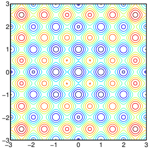
- ill-conditioning

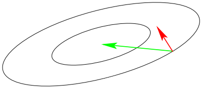
  widely varying sensitivity

- ruggedness

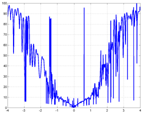
  non-smooth, discontinuous, multimodal, and/or noisy function







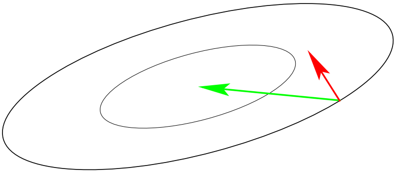
gradient direction Newton direction



# Ill-Conditioned Problems

Curvature of level sets

Consider the convex-quadratic function $f(\underline{x}) = \frac{1}{2}(\underline{x} - \underline{x}^*)^T \underline{\underline{H}}(\underline{x} - \underline{x}^*)$



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

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

Condition number equals nine here. Condition numbers between $100$ and even $10^{10}$ can often be observed in real world problems.

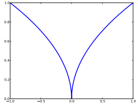
If $\underline{\underline{H}} \approx \underline{\underline{I}}$ (small condition number of $\underline{\underline{H}}$) first order information (e.g. the gradient) is sufficient. Otherwise **second order information** (estimation of $\underline{\underline{H}}^{-1}$) **is required.**

# What Makes a Function Difficult to Solve?

Why stochastic search?

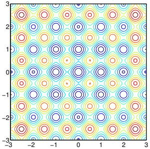
- non-linear, non-quadratic, non-convex

  on linear/quadratic functions better search policies are available



- dimensionality

  (considerably) larger than three

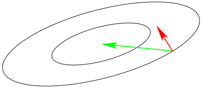


- non-separability

  dependencies between the objective variables

- ill-conditioning

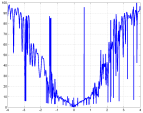
  widely varying sensitivity



gradient direction Newton direction

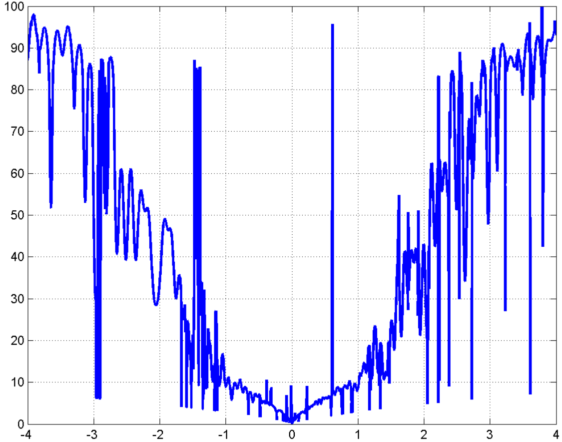
- ruggedness

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



# Ruggedness

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




cut from an (easily) solvable example in 5-D

# What Makes a Function Difficult to Solve?

...and what can be done

| Challenge | Approach in Evolutionary Computation |
|---|---|
| Dimensionality, Non-Separability | exploiting the problem structure<br>locality, neighborhood, encoding |
| Ill-conditioning | second order approach<br>changes the neighborhood metric |
| Ruggedness | **non-local** policy, large sampling width (step-size)<br>as large as possible while preserving a reasonable convergence speed |
| | stochastic, non-elitistic, **population-based** method<br>recombination operator<br>serves as repair mechanism |

. . . metaphores

# Metaphors

| (Biological) **Evolution**(ary Computation) | | **Optimization** |
|---|---|---|
| genome | $\longleftrightarrow$ | decision variables |
| | | design variables |
| | | object variables |
| individual, offspring, parent | $\longleftrightarrow$ | candidate solution |
| population | $\longleftrightarrow$ | set of candidate solutions |
| fitness function | $\longleftrightarrow$ | objective function |
| | | loss function |
| | | cost function |
| generation | $\longleftrightarrow$ | iteration |

# Stochastic Search

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

**Initialize distribution parameters $\underline{\theta}$, set population size $\lambda \in \mathbb{N}$**
**While not terminate**

1. **Sample distribution** $P\left(\underline{x}|\underline{\theta}\right) \to \underline{x}_1, \ldots, \underline{x}_\lambda \in \mathbb{R}^n$
2. **Evaluate** $\underline{x}_1, \ldots, \underline{x}_\lambda$ **on** $f$
3. **Update parameters** $\underline{\theta} \leftarrow F_\theta(\underline{\theta}, \underline{x}_1, \ldots, \underline{x}_\lambda, f(\underline{x}_1), \ldots, f(\underline{x}_\lambda))$

Everything depends on the definition of $P$ and $F_\theta$

deterministic algorithms are covered as well

In Evolutionary Algorithms the distribution $P$ is often implicitly defined via **operators on a population**, in particular, selection, recombination and mutation

Natural template for *Estimation of Distribution Algorithms*

# Stochastic Search

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

**Initialize distribution parameters $\underline{\theta}$, set population size $\lambda \in \mathbb{N}$**

**While not terminate**

1. **Sample distribution** $P(\underline{x}|\underline{\theta}) \to \underline{x}_1, \ldots, \underline{x}_\lambda \in \mathbb{R}^n$
2. **Evaluate** $\underline{x}_1, \ldots, \underline{x}_\lambda$ **on** $f$
3. **Update parameters** $\underline{\theta} \leftarrow F_\theta(\underline{\theta}, \underline{x}_1, \ldots, \underline{x}_\lambda, f(\underline{x}_1), \ldots, f(\underline{x}_\lambda))$

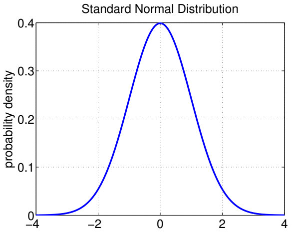
## In the following

- $P$ is a **multi-variate normal** distribution $\mathcal{N}\left(\underline{m}, \sigma^2\underline{\underline{C}}\right) \sim \underline{m} + \sigma\mathcal{N}\left(\underline{0}, \underline{\underline{C}}\right)$

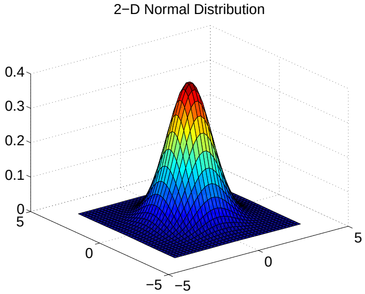
  $\underline{\theta} = \{\underline{m}, \underline{\underline{C}}, \sigma\} \in \mathbb{R}^n \times \mathbb{R}^{n\times n} \times \mathbb{R}_+$
- $F_\theta = F_\theta(\underline{\theta}, \underline{x}_{1:\lambda}, \ldots, \underline{x}_{\mu:\lambda})$, where $\mu \leq \lambda$ and $\underline{x}_{i:\lambda}$ is the $i$-th best of the $\lambda$ points

. . . normal distribution

# Normal Distribution




probability density of 1-D standard normal distribution



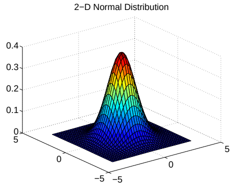

probability density of 2-D normal distribution

# The Multi-Variate ($n$-Dimensional) Normal Distribution

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

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

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

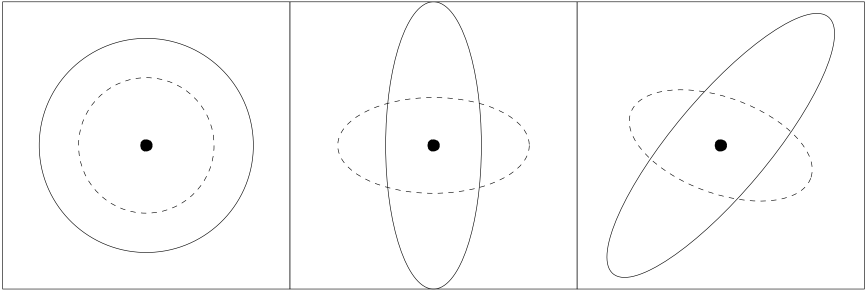



The **covariance matrix** $\underline{\underline{C}}$

- determines the shape
- **geometrical interpretation**: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\underline{x} \in \mathbb{R}^n \mid \underline{x}^{\mathrm{T}} \underline{\underline{C}}^{-1} \underline{x} = 1\}$

…any **covariance matrix** can be uniquely identified with the iso-density ellipsoid $\{\underline{x} \in \mathbb{R}^n \,|\, \underline{x}^{\mathrm{T}} \underline{\underline{C}}^{-1} \underline{x} = 1\}$

Lines of Equal Density



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

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

…CMA

# Rank-Based Stochastic Search

**Rank-based black box search to minimize $f : \mathbb{R}^n \to \mathbb{R}$**

**Initialize distribution parameters $\underline{\theta}$, set population size $\lambda \in \mathbb{N}$**

**While not terminate**

1. **Sample distribution** $P\left(\underline{x}|\underline{\theta}\right) \to \underline{x}_1, \ldots, \underline{x}_\lambda \in \mathbb{R}^n$
2. **Evaluate** $\underline{x}_1, \ldots, \underline{x}_\lambda$ **on** $f$

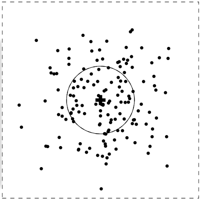
   **and let** $f(\underline{x}_{i:\lambda}) \leq f(\underline{x}_{j:\lambda}) \Leftrightarrow i \leq j$
3. **Update parameters** $\underline{\theta} \leftarrow F_\theta(\underline{\theta}, \underline{x}_{1:\lambda}, \ldots, \underline{x}_{\mu:\lambda})$

$P$ is a multi-variate normal distribution $\mathcal{N}\left(\underline{m}, \sigma^2 \underline{\underline{C}}\right) \sim \underline{m} + \sigma \mathcal{N}\left(\underline{0}, \underline{\underline{C}}\right)$

. . . sampling

# Sampling New Search Points

## The Mutation Operator



**New search points are normally distributed**

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

perturbations of $\underline{m}$

where

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

The question remains how to update $\underline{m}$, $\sigma$, and $\underline{\underline{C}}$.

# Update of the Distribution Mean $\underline{m}$

Selection and Recombination

**The best** $\mu$ **points** are selected **from the sampled solutions** (non-elitistic) and a **weighted mean** is taken:

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

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

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

where

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

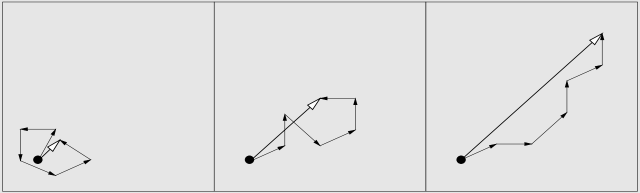
# Path Length Control

## The Concept

$$\underline{x}_i = \underline{m} + \sigma\, \underline{y}_i$$
$$\underline{m} \leftarrow \underline{m} + \sigma \underline{y}_w$$

### Measure the length of the *evolution path*

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



$\Downarrow$ decrease $\sigma$ | $\Downarrow$ increase $\sigma$

loosely speaking steps are

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

# Path Length Control: Equations

AKA Cumulative Step-Size Adaptation (CSA)

Initialize $\underline{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\underline{\underline{C}} = \underline{\underline{I}}$, and $\underline{p}_\sigma = \underline{0}$
set $c_\sigma \approx 4/n$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
set $\lambda$ and $w_{i=1,\dots,\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \approx 0.3\,\lambda$

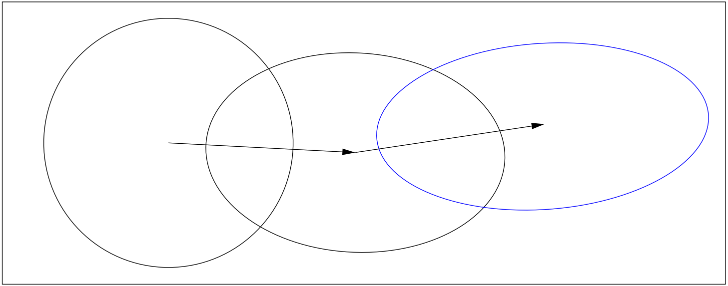
While not terminate

$$\begin{array}{lcll}
\underline{x}_i & = & \underline{m} + \sigma\,\underline{y}_i, \quad \text{where } \underline{y}_i \sim \mathcal{N}_i\left(\underline{0}, \underline{\underline{C}}\right) \text{ for } i = 1, \dots, \lambda & \text{sampling} \\
\underline{m} & \leftarrow & \underline{m} + \sigma \underline{y}_w \quad \text{where } \underline{y}_w = \sum_{i=1}^{\mu} w_i\, \underline{y}_{i:\lambda} & \text{update mean} \\
\underline{p}_\sigma & \leftarrow & (1 - c_\sigma)\, \underline{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \underline{\underline{C}}^{-\frac{1}{2}} \underline{y}_w & \text{cumulation for } \sigma \\
\sigma & \leftarrow & \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\underline{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\underline{0}, \underline{\underline{I}})\|} - 1\right)\right) & \text{update of } \sigma
\end{array}$$

# Covariance Matrix Adaptation

## Rank-One Update

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



new distribution,

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

the ruling principle: the adaptation **increases the likelyhood of successful steps**, $\underline{y}_w$, to appear again

...equations

# Preliminary Set of Equations

## Covariance Matrix Adaptation with Rank-One Update

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

$$
\begin{aligned}
\underline{x}_i &= \underline{m} + \sigma\, \underline{y}_i, \qquad \underline{y}_i \sim \mathcal{N}_i\left(\underline{0}, \underline{\underline{C}}\right), \qquad i = 1, \dots, \lambda \\
\underline{m} &\leftarrow \underline{m} + \sigma \underline{y}_w \qquad \text{where } \underline{y}_w = \sum_{i=1}^{\mu} w_i\, \underline{y}_{i:\lambda} \\
\underline{\underline{C}} &\leftarrow (1 - c_{\text{cov}})\underline{\underline{C}} + c_{\text{cov}} \mu_w \underbrace{\underline{y}_w \underline{y}_w^{\mathrm{T}}}_{\text{rank-one}} \qquad \text{where } \mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \geq 1
\end{aligned}
$$

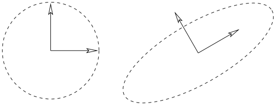
$\lambda$ can be small

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

The 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 $\underline{y}_w$, sequentially in time and space
  eigenvectors of the covariance matrix $\underline{\underline{C}}$ are the principle components / the principle axes of the mutation ellipsoid



- learns a new, **rotated problem representation** and a **new variable metric** (Mahalanobis)
  components are independent (only) in the new representation
  **rotational invariant**
  equivalent with an adaptive (general) linear encoding

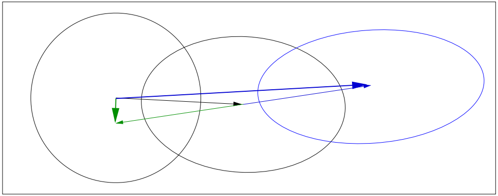
- approximates the **inverse Hessian** on convex-quadratic functions
  overwhelming empirical evidence, proof is in progress

...contents

# Cumulation

## Utilizing the Evolution Path

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



$$\underline{p}_{\mathrm{c}} \leftarrow \underbrace{(1 - c_{\mathrm{c}})}_{\text{decay factor}} \underline{p}_{\mathrm{c}} + \underbrace{\sqrt{1 - (1 - c_{\mathrm{c}})^2} \sqrt{\mu_w}}_{\text{normalization factor}} \underline{y}_w$$

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

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

... $\mathcal{O}(n^2)$ 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

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

... rank-$\mu$

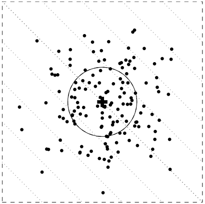
# Rank-$\mu$ Update

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

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

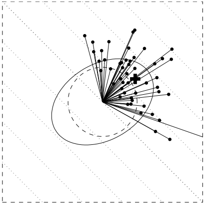
$$\underline{\underline{C}} \leftarrow (1 - c_{\mathrm{cov}})\, \underline{\underline{C}} + c_{\mathrm{cov}} \sum_{i=1}^{\mu} w_i\, \underline{y}_{i:\lambda} \underline{y}_{i:\lambda}^{\mathrm{T}}$$

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



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

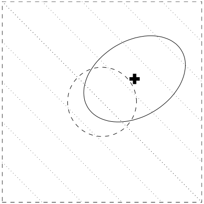
sampling of $\lambda = 150$ solutions where $\underline{\underline{C}} = \underline{\underline{I}}$ and $\sigma = 1$



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

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

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



$\underline{m}_{\text{new}} \leftarrow \underline{m} + \frac{1}{\mu} \sum \underline{y}_{i:\lambda}$

new distribution

The **rank-$\mu$ update**

- increases the possible learning rate in large populations
  roughly from $2/n^2$ to $\mu_w/n^2$
- can reduce the number of necessary **generations** roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$[5]
  given $\mu_w \propto \lambda \propto n$

Therefore the rank-$\mu$ update is the primary mechanism whenever a large population size is used

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

[5] 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

# Summary of Equations

The Covariance Matrix Adaptation Evolution Strategy

Input: $\underline{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\lambda$
Initialize $\underline{\underline{C}} = \underline{\underline{I}}$, and $\underline{p}_c = \underline{0}$, $\underline{p}_\sigma = \underline{0}$,
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

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

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

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

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

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

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

# Experimentum Crucis

What did we want to achieve?

- reduce any convex-quadratic function

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

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

to the sphere model

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

without use of derivatives

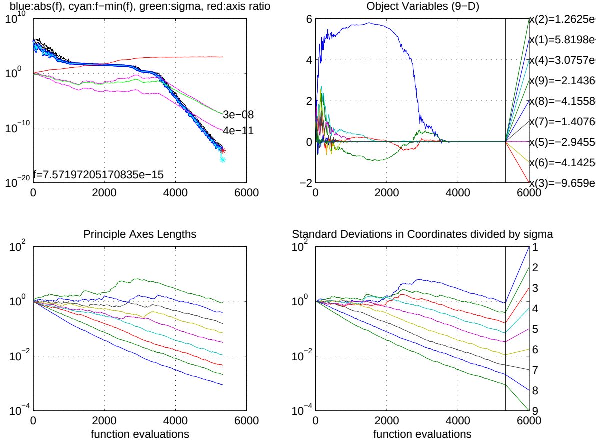
- lines of equal density align with lines of equal fitness

$$\underline{\underline{C}} \propto \underline{\underline{H}}^{-1}$$

in a stochastic sense

# Experimentum Crucis (1)

$f$ convex-quadratic, separable

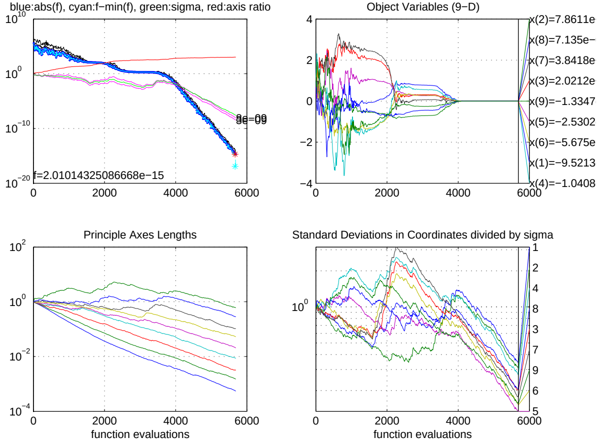



$$f(\underline{x}) = \sum_{i=1}^{n} 10^{\alpha \frac{i-1}{n-1}} x_i^2, \alpha = 6$$

. . . crucis rotated

# Experimentum Crucis (2)

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




$\underline{C} \propto \underline{\underline{H}}^{-1}$ **for all** $g, \underline{\underline{H}}$

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

. . . on convergence

# On Convergence

- convergence on a very **broad class of functions**, e.g. for Monte Carlo pure random search

very slow: $\|\underline{x} - \underline{x}^*\| \propto \frac{1}{t^{1/n}} = \frac{1}{\exp\left(\frac{\log t}{n}\right)}$

- convergence with practically **feasible convergence rates** on, e.g., $g\left(\frac{1}{2}\underline{x}^{\mathrm{T}}\underline{\underline{H}}\underline{x}\right)$

CMA-ES $\Rightarrow \|\underline{x} - \underline{x}^*\| \propto \frac{1}{\exp\left(\frac{t/4}{n}\right)}$

# Evaluation/Selection of Search Algorithms

Evaluation (of the performance) of a search algorithm needs

- meaningful quantitative measure on benchmark functions or real world problems
- account for meta-parameter tuning

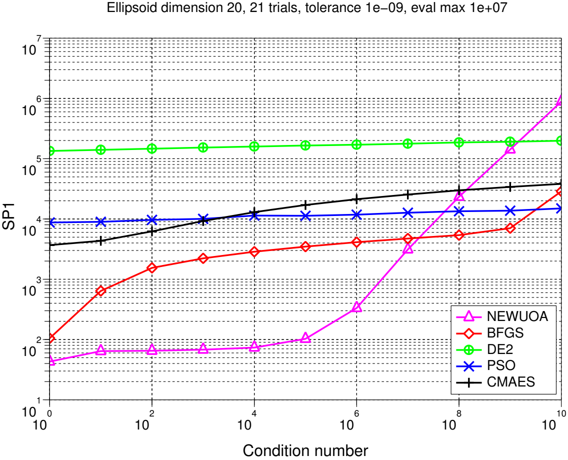
  can be quite expensive
- account for invariance properties (symmetries)

  prediction of performance is based on "similarity", ideally equivalence classes of functions
- account for algorithm internal cost

  often negligible, depending on the objective function cost

# Comparison to BFGS, NEWUOA, PSO and DE (1)

$f$ convex-quadratic, separable with varying $\alpha$



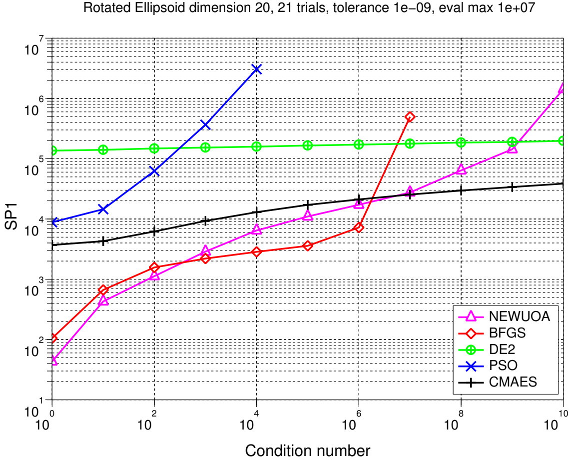

$f(\underline{x}) = g(\underline{x}^{\mathrm{T}}\underline{\underline{H}}\underline{x})$ with
$g$ identity (BFGS, NEWUOA) or
$g$ order-preserving (strictly increasing, all other)

SP1 = average number of objective function evaluations to reach the target function value of $10^{-9}$

. . . population size, invariance

# Comparison to BFGS, NEWUOA, PSO and DE (2)

$f$ convex-quadratic, non-separable (rotated) with varying $\alpha$




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

$g$ identity (BFGS, NEWUOA) or

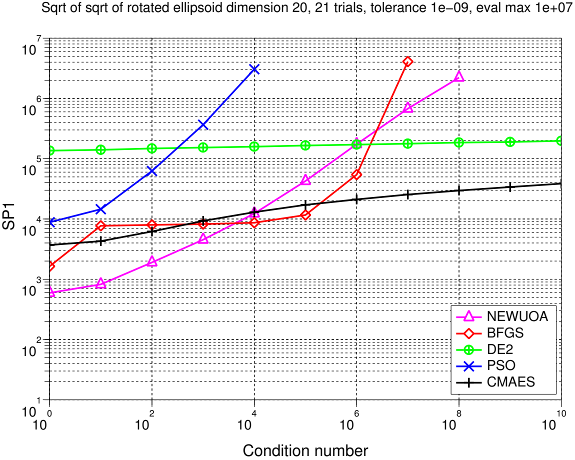
$g$ order-preserving (strictly increasing, all other)

SP1 = average number of objective function evaluations to reach the target function value of $10^{-9}$

. . . population size, invariance

# Comparison to BFGS, NEWUOA, PSO and DE (3)

$f$ non-convex, non-separable (rotated) with varying $\alpha$




$f(\underline{x}) = g(\underline{x}^T \underline{\underline{H}} x)$ with

$g(.) = (.)^{1/4}$ (BFGS, NEWUOA) or

$g$ any order-preserving (strictly increasing, all other)

SP1 = average number of objective function evaluations to reach the target function value of $10^{-9}$

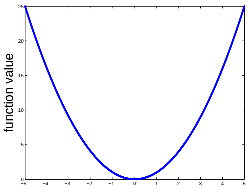
. . . population size, invariance

# Invariance

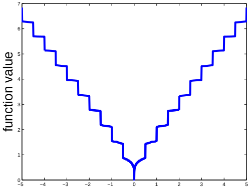
## The short version

*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.*

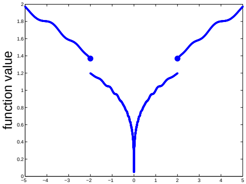
— Albert Einstein




$$f(x) = x^2$$




$$f(x) = g_1(x^2)$$




$$f(x) = g_2(x^2)$$

- all three functions are **equivalent** for rank-based search methods

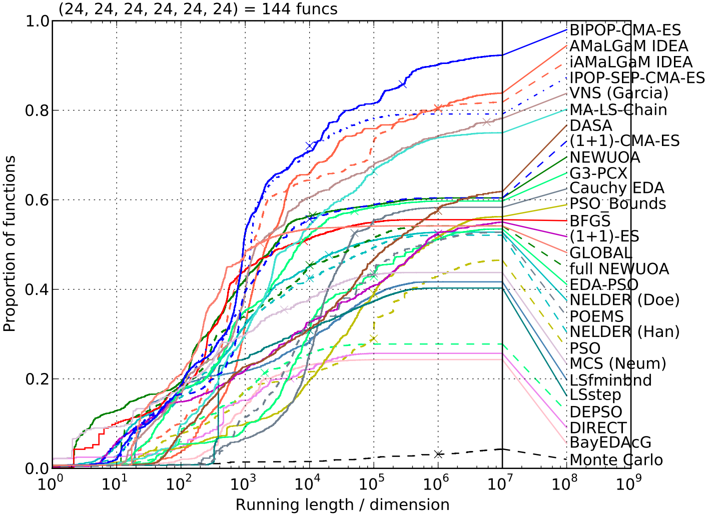
  large equivalence class

- invariance allows a save **generalization** of empirical results

  here on $f(x) = x^2$ (left) to any $f(x) = g(x^2)$, where $g$ is monotonous

# Comprehensive Comparison of 28 Algorithms

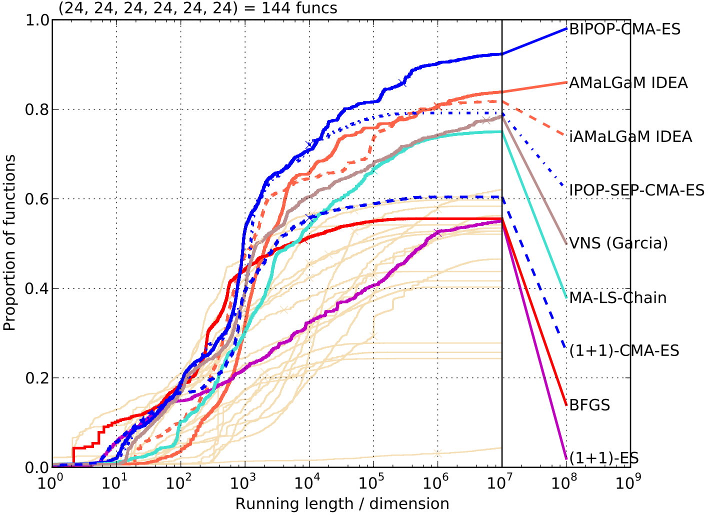
Empirical Distribution of Expected Running Length




on 24 benchmark functions of BBOB 2009 in 20-D

# Comprehensive Comparison of 28 Algorithms

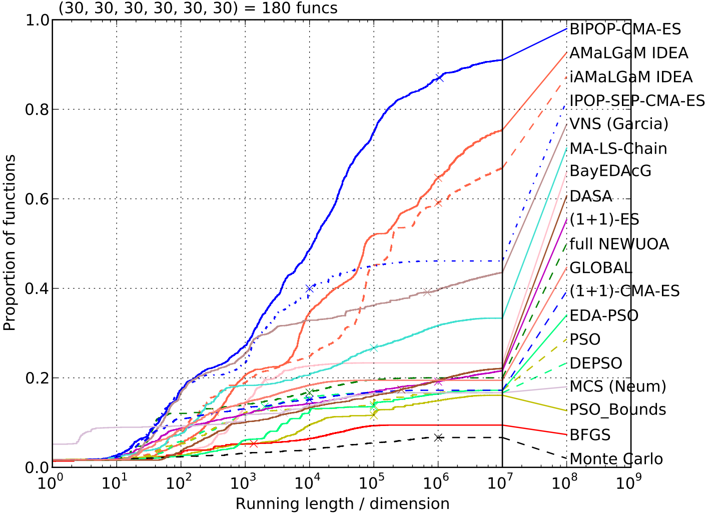
Empirical Distribution of Expected Running Length




on 24 benchmark functions of BBOB 2009 in 20-D

# Comprehensive Comparison of 19 Algorithms

Empirical Distribution of Expected Running Length




on 30 noisy benchmark functions of BBOB 2009 in 20-D

# Summary

Covariance Matrix Adaptation Evolution Strategy (CMA-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

   preserves invariance

3. *Covariance matrix adaptation (CMA)* **increases the likelyhood of previously successful steps**

   learning all pairwise dependencies
   $\Longrightarrow$ adapts a variable metric
   $\Longrightarrow$ new (rotated) problem representation

4. An **evolution path** (a non-local trajectory)
   - enhances the covariance matrix (rank-one) adaptation

     yields sometimes linear time complexity
   - controls the **step-size** (step length)

     aims at conjugate perpendicularity

# Merci !

http://www.lri.fr/~hansen/cmaesintro.html
or google NIKOLAUS HANSEN