CMA-ES – a Stochastic Second-Order Method for Function-Value Free Numerical Optimization

Nikolaus Hansen
INRIA, Research Centre Saclay
Machine Learning and Optimization Team, TAO
Univ. Paris-Sud, LRI

MSRC October-December 2011

http://www.inria.fr

http://www.lri.fr/~hansen

INRIA: The French National Institute for Research in Computer science and Control
8 research centres, 210 research teams

Outline

- introduction: continuous black-box optimization
  - a mathematical formulation
  - applications of black-box optimization
  - why is it difficult?
- a stochastic second-order black-box optimization method ”from first principles”: CMA-ES
- on design principles (invariance and others)
- experimental results

...feel free to ask questions...

Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius, and a lot of courage, to move in the opposite direction.

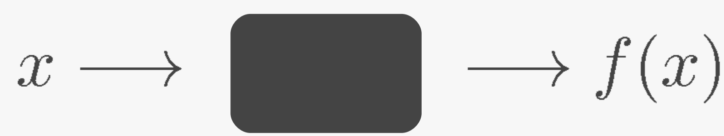
Albert Einstein

Black-Box Optimization (Search)

Minimize (or maximize) a given continuous domain objective (cost, loss, error, fitness) function

$$f : \mathbb{R}^n \to \mathbb{R}, \quad x \mapsto f(x)$$

where f is considered as a black-box



and in particular

- gradients are not cheaply available or useful
- problem specific knowledge is used *within* the black box, e.g. with an appropriate encoding

and typically $2 \leq n \leq 200$

Objective: find $x \in \mathbb{R}^n$ with small $f(x)$, where the search costs are the number of back-box calls (function evaluations)

Typical Applications

- model/system calibration
  - biological/chemical/physical $\Longrightarrow$ universal constants
  - production process
- optimization of control parameters
  - movements of a robot
  - trajectory of a rocket
  - stability of a gas flame
- shape optimization
  - curve fitting
  - aero- or fluid dynamics design (airfoil, airship)

On-line registration of spline images

Intraoperative ultrasound image CT image

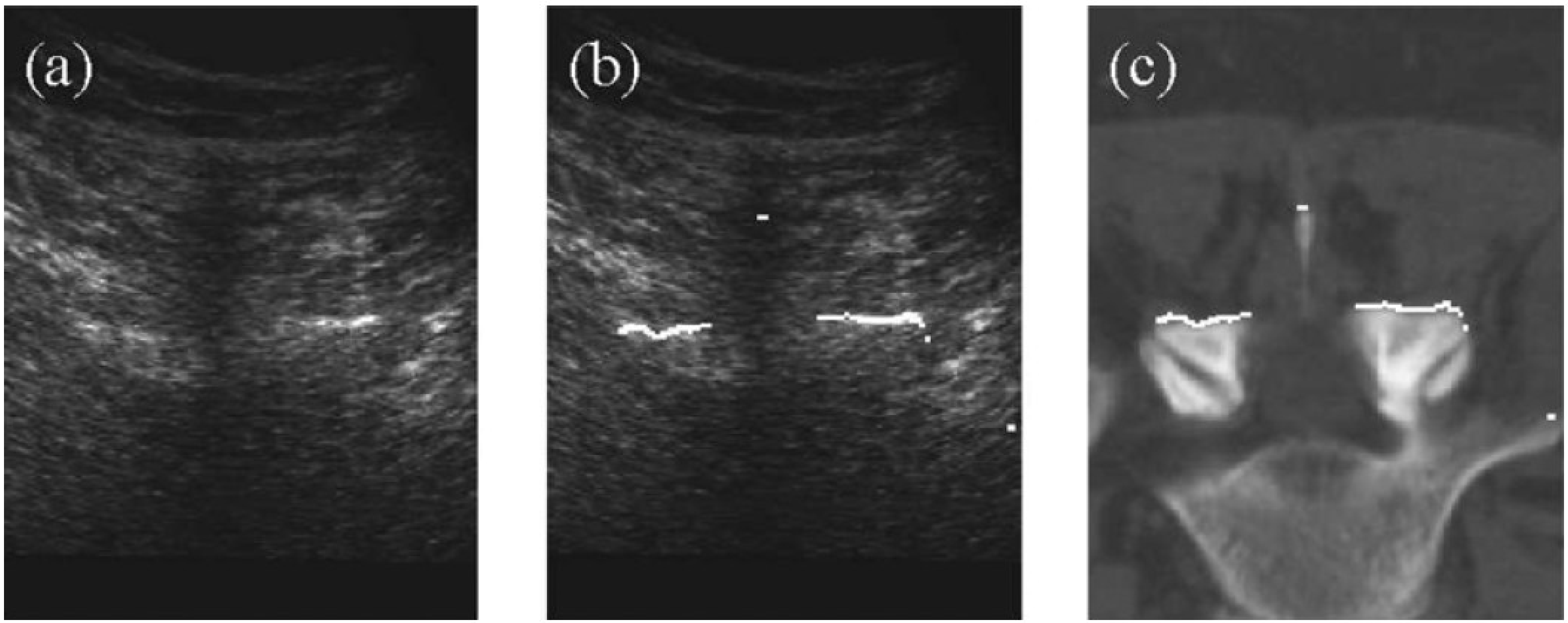


Fig. 6. (a) Intraoperative axial ultrasound image of a vertebra. (b) Bone surface at the registered position. (c) Corresponding CT image.

from [Winter et al 2008,

Registration of CT and intraoperative 3-D ultrasound images of the spine using evolutionary and gradient-based methods]

Distribution of final misalignment

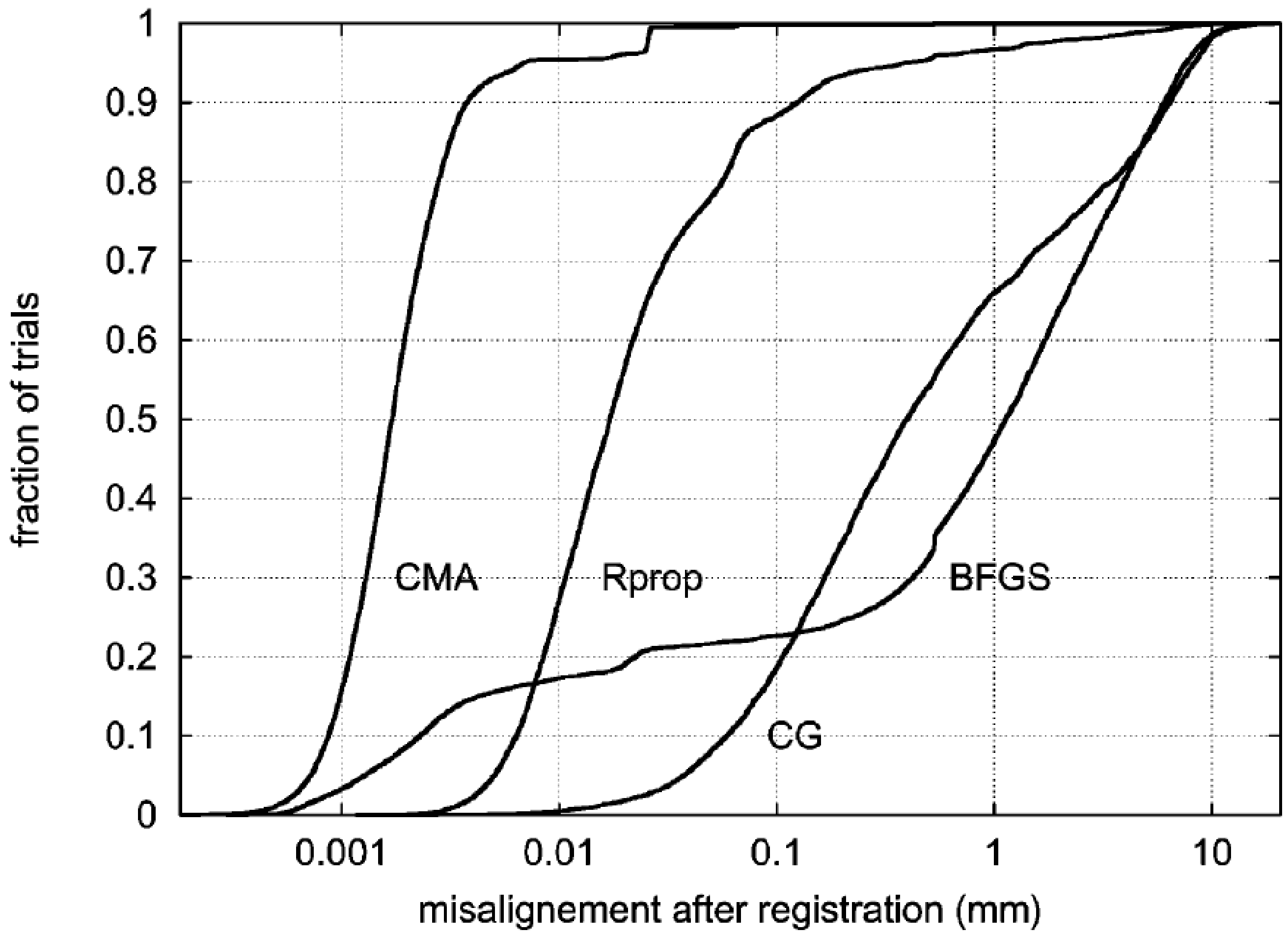
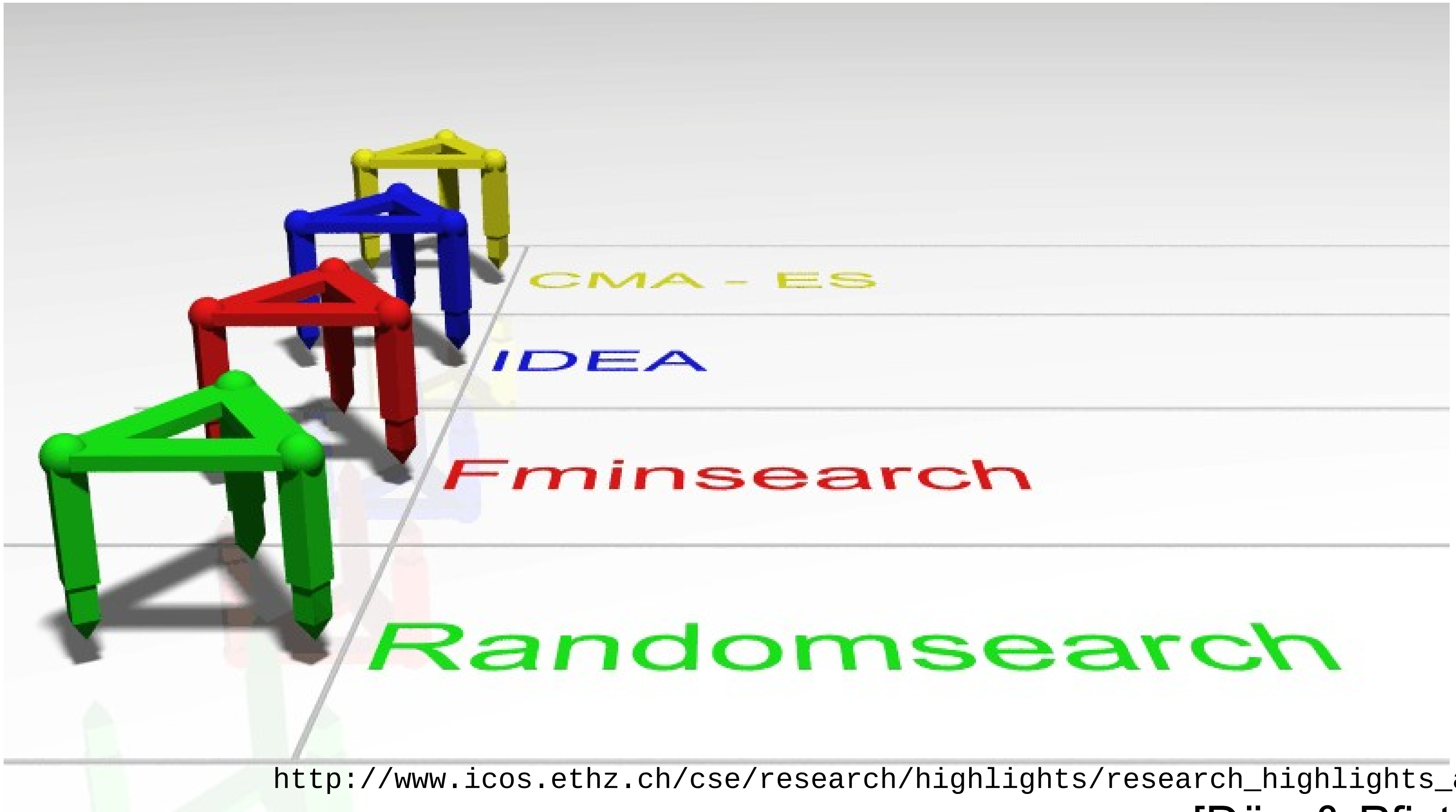


Fig. 9. Misalignment of all registration trials in the multistart optimization scenario; registration of 12 vertebrae, each with 1000 different starting positions and the different optimization methods BFGS, CG, iRprop, CMA.

from [Winter et al 2008]

Optimization of walking gaits



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

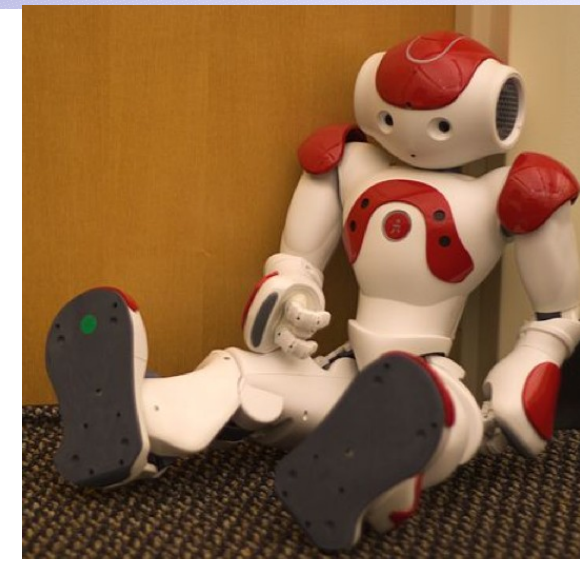
[Dürr & Pfister 2004]

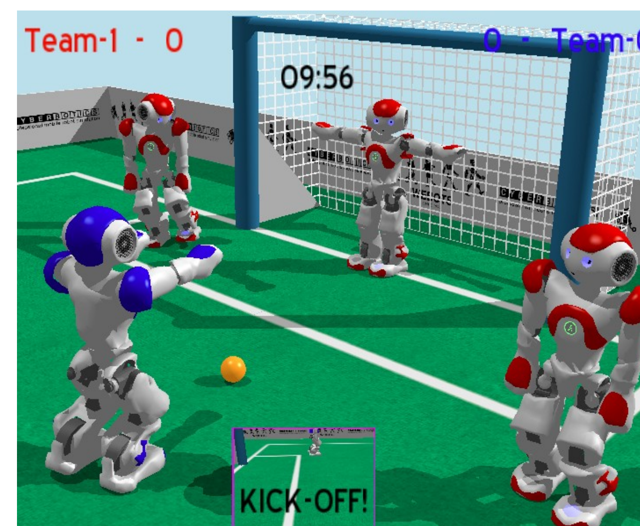
CMA-ES, Covariance Matrix Adaptation Evolution Strategy [Hansen et al 2003]
IDEA, Iterated Density Estimation Evolutionary Algorithm [Bosman 2003]
Fminsearch, downhill simplex method [Nelder & Mead 1965]

RoboCup 3D Simulated Soccer League

Competition based on the Nao robot, in the RoboCup 2011:

- Team UT Austin Villa, University of Texas at Austin, won amongst 22 teams
- UT Austin Villa won all 24 games scoring 136 goals and conceding none (!)
- This was largely because of the agent's superior skills, optimized using CMA-ES



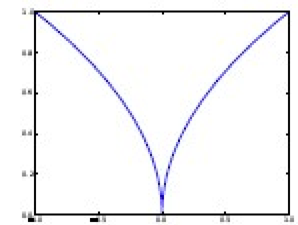


2011 Final

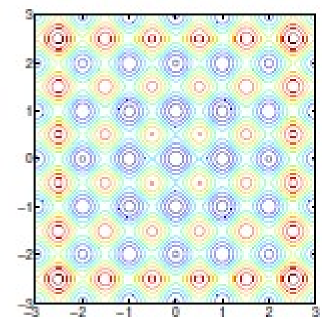
http://www.youtube.com/watch?v=DmyHUMt2U0w

Difficulties in black-box optimization

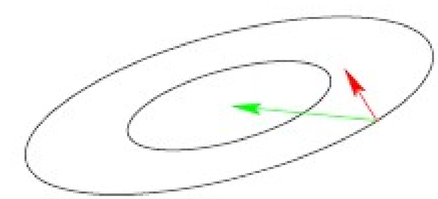
- non-linear, non-quadratic, non-convex
  on linear/quadratic functions better search policies are available



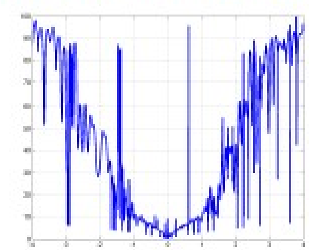
- dimensionality
  (considerably) larger than three



- non-separability
  dependencies between the objective variables
- ill-conditioning
  widely varying sensitivity



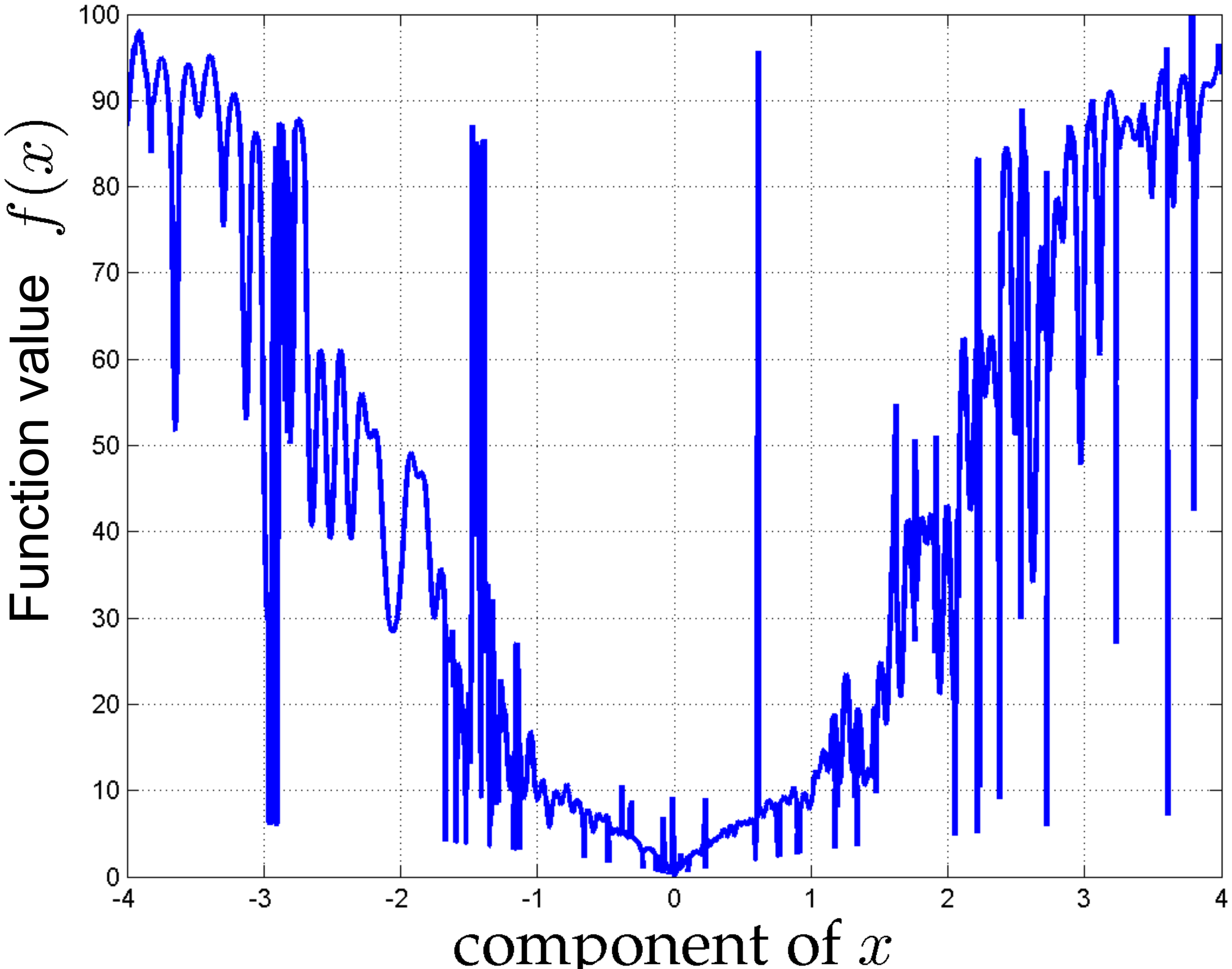
- ruggedness
  non-smooth, discontinuous, multimodal, and/or noisy function



in any case the objective function must be highly regular

Rugged landscape

Section through 5-D ($n = 5$) landscape



$$f : \mathbb{R}^n \to \mathbb{R}, \quad x \mapsto f(x)$$

The Methods

Taxonomy of search methods

Gradient-based methods (Taylor, smooth)

local search

- Conjugate gradient methods [Fletcher & Reeves 1964]
- Quasi-Newton methods (BFGS) [Broyden et al 1970]

Derivative-free optimization (DFO)

- Trust-region methods (NEWUOA) [Powell 2006]
- Simplex downhill [Nelder & Mead 1965]
- Pattern search [Hooke & Jeeves 1961] [Audet & Dennis 2006]

Stochastic (randomized) search methods

- Evolutionary algorithms [Rechenberg 1965, Holland 1975]
- Simulated annealing (SA) [Kirkpatrick et al 1983]
- Simultaneous perturbation stochastic approximation (SPSA)[Spall 2000]

Taxonomy of Evolutionary Algorithms

- Genetic Algorithms
  - operate on bit-strings
  - operators (recombination mutation) are typically problem-tailored
- Evolution Strategies, Evolutionary Programming, Differential Evolution, Particle Swarm Optimization
  - operate on continuous search spaces
  - with problem-tailored encoding of the fitness
- Genetic Programming
  - operates on trees
  - evolving computer programs

Metaphores

Evolutionary Computation		Optimization
individual, offspring, parent	$\longleftrightarrow$	candidate solution decision variables design variables object variables
population	$\longleftrightarrow$	set of candidate solutions
fitness function	$\longleftrightarrow$	objective function loss function cost function error function
generation	$\longleftrightarrow$	iteration

...a stochastic optimization method...

Stochastic optimization template

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

While not *happy*

1. Sample $P(x|\theta) \to x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on $f \longrightarrow f(x_1), \ldots, f(x_\lambda)$
3. Update parameters
   $\theta \leftarrow Update(\theta, x_1, \ldots, x_\lambda, \widehat{w}(f(x_1)), \ldots, \widehat{w}(f(x_\lambda)) \in \mathbb{R}^\lambda)$

Return, e.g., the expected value of P: $m \in \theta$

Stochastic optimization template

Initialize parameters θ, set population size $\lambda \in \mathbb{N}$
While not *happy*

1. Sample $P(x|\theta) \to x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on $f \longrightarrow f(x_1), \ldots, f(x_\lambda)$
3. Update parameters
   $\theta \leftarrow Update(\theta, x_1, \ldots, x_\lambda, \widehat{w}(f(x_1)), \ldots, \widehat{w}(f(x_\lambda)) \in \mathbb{R}^\lambda)$

Return, e.g., the expected value of P: $m \in \theta$

Remark: we replaced searching for $\arg\min_x f(x)$ with searching for $\arg\min_\theta Ew(f(x))$

Crucial questions: how to choose the *parametrized* distribution P, the objective $\widehat{w}$ and the update function $Update$

Instantiation of the optimization template

- choose maximum entropy distribution

  in continuous domain, $f : \mathbb{R}^n \to \mathbb{R}$,

  the multivariate normal distribution

- use only the ranking of solutions in the update

  determines $\widehat{w}$ up to a monotonous shaping function

- apply a natural gradient descent to the distribution parameters

  maximal improvement of the "expected fitness"

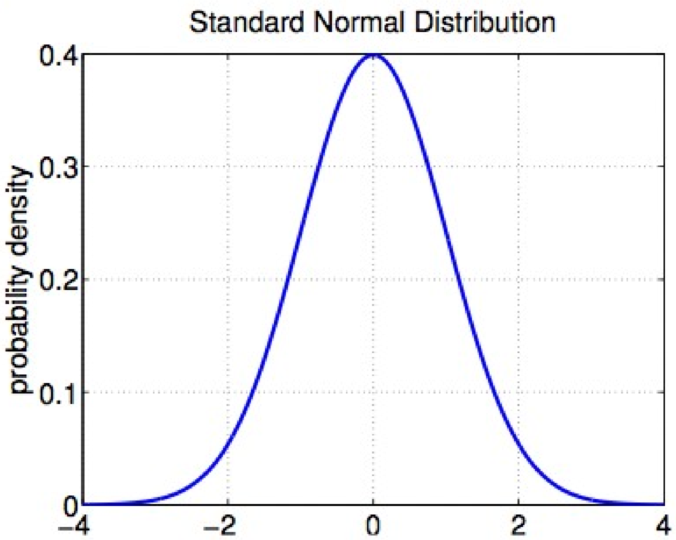
  [Wiestra et al 2008, Akimito et al 2010, Glasmachers et al 2010, Malagò et al 2011, Arnold et al 2011]

- use additional non-local information (low pass filtering) and strive for conjugate-orthogonal steps

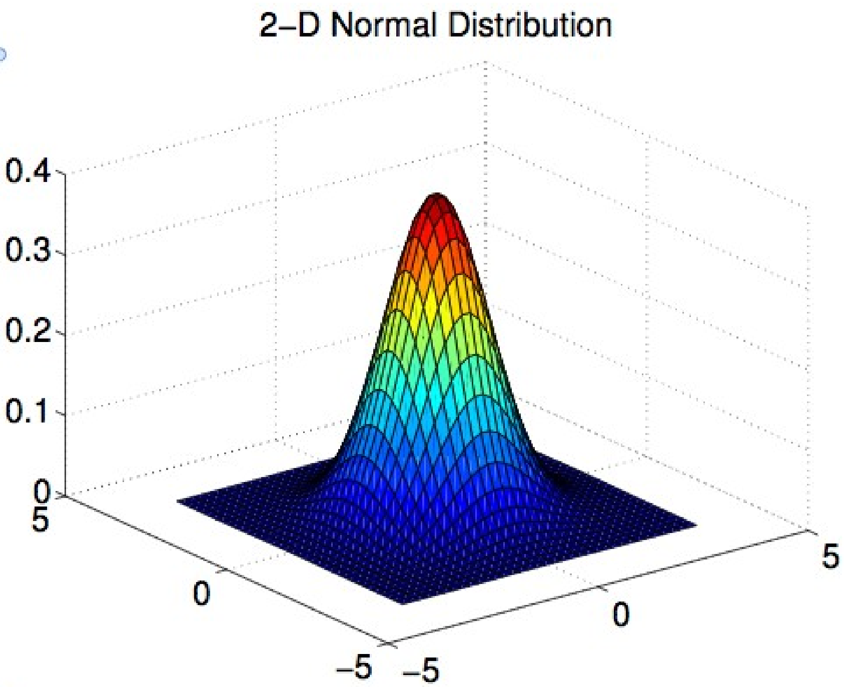
  two well-known "tricks"

$\Longrightarrow$ CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [Hansen&Ostermeier 1996, 2001, Hansen et al 2003, Hansen&Kern 2004, Jastrebski&Arnold 2006]

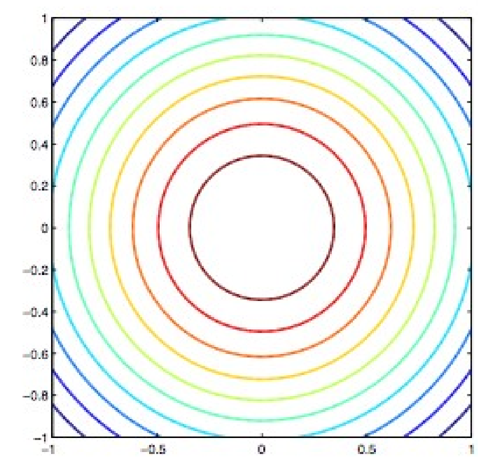
Normal (Gaussian) Distribution



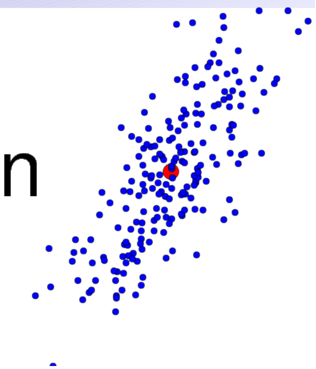
probability density of the 1-D standard normal distribution



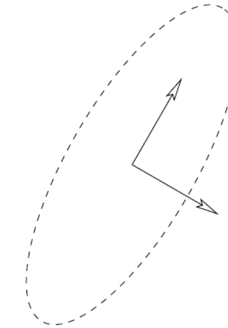
probability density of a 2-D normal distribution



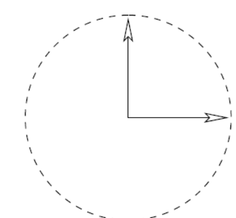
Sample distribution in continuous domain, $f : \mathbb{R}^n \to \mathbb{R}$, is a full multivariate normal distribution



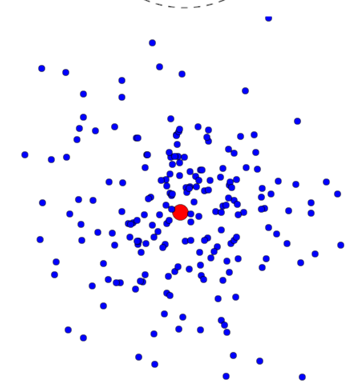
$$\mathcal{N}(m, \mathbf{C}) \in \mathbb{R}^n \qquad \theta = (m, \mathbf{C})$$



- mean $m \in \mathbb{R}^n$ is the prototyp solution,
- covariance matrix $\mathbf{C}$ determines variations around m, we have

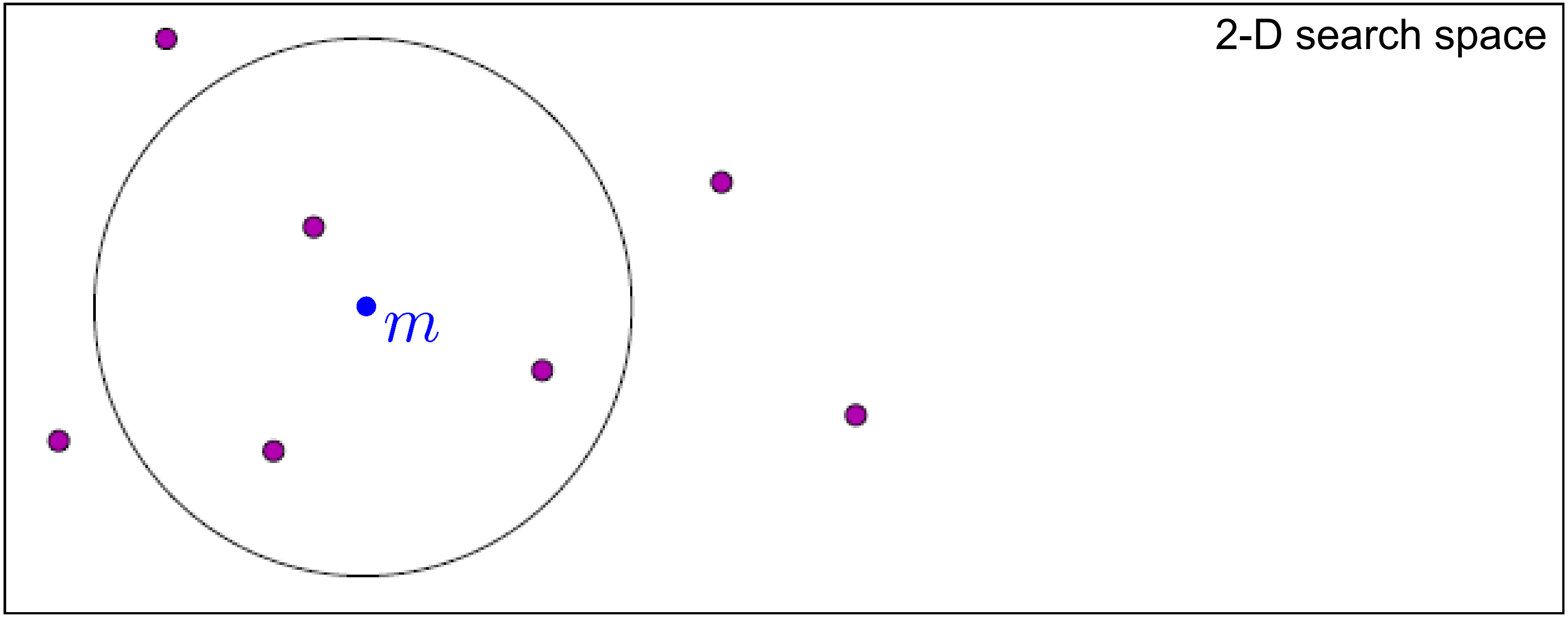


$$\mathcal{N}(m, \mathbf{C}) = m + \mathcal{N}(\mathbf{0}, \mathbf{C})$$



The density is $\propto \exp^{1/2}\left(-\|\mathbf{x} - m\|^2_{\mathbf{C}^{-1}}\right)$
$= \exp^{1/2}\left(-(\mathbf{x} - m)\mathbf{C}^{-1}(\mathbf{x} - m)\right)$

Covariance Matrix Adaptation

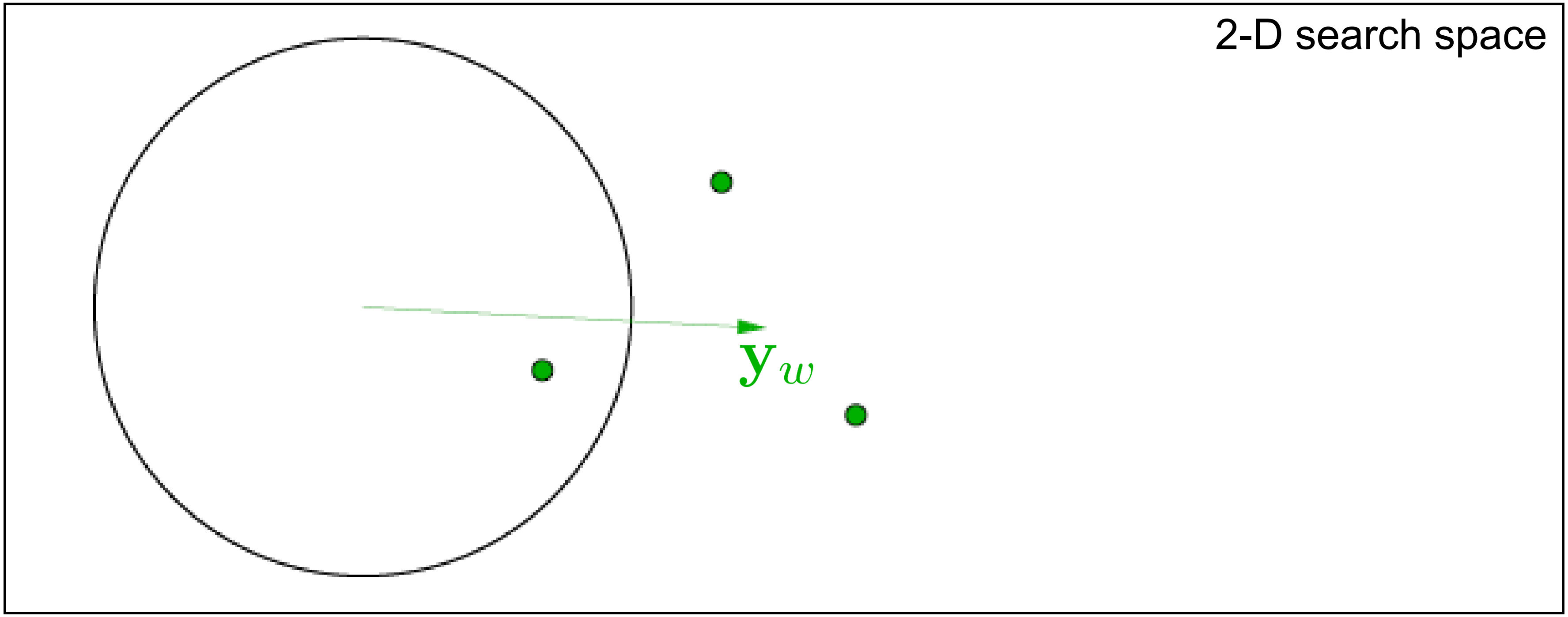


sample from the initial distribution: $\mathcal{N}(m, \mathbf{I})$, $\mathbf{C} = \mathbf{I}$

.

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

Covariance Matrix Adaptation

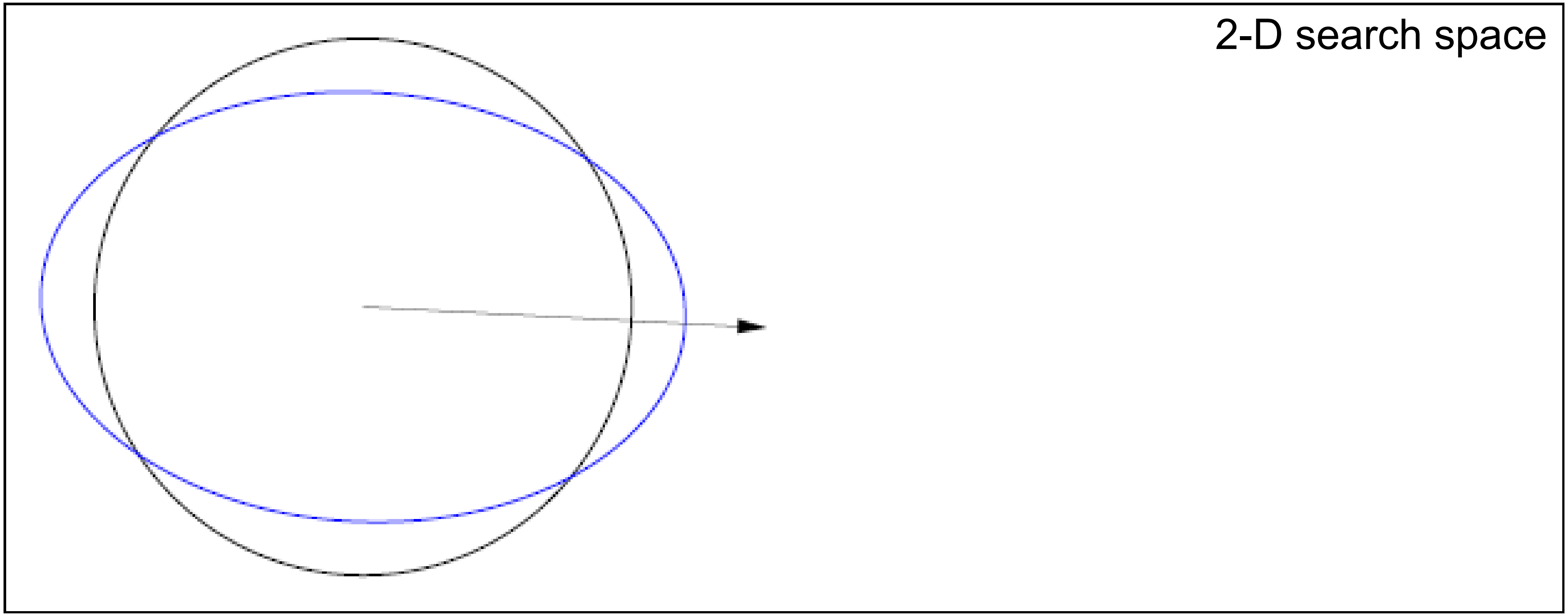


$\mathbf{y}_w$ is the move of the mean m, disregarding σ .

.

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

Covariance Matrix Adaptation



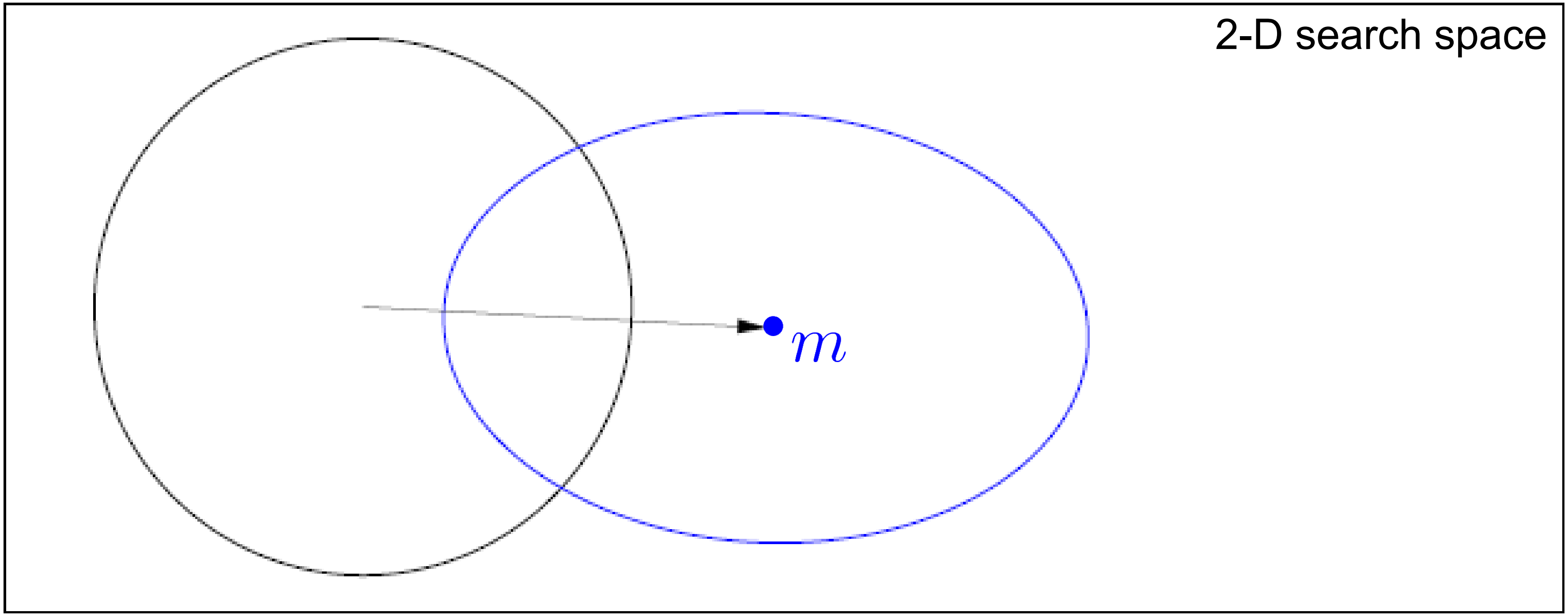
mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

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

.

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

Covariance Matrix Adaptation

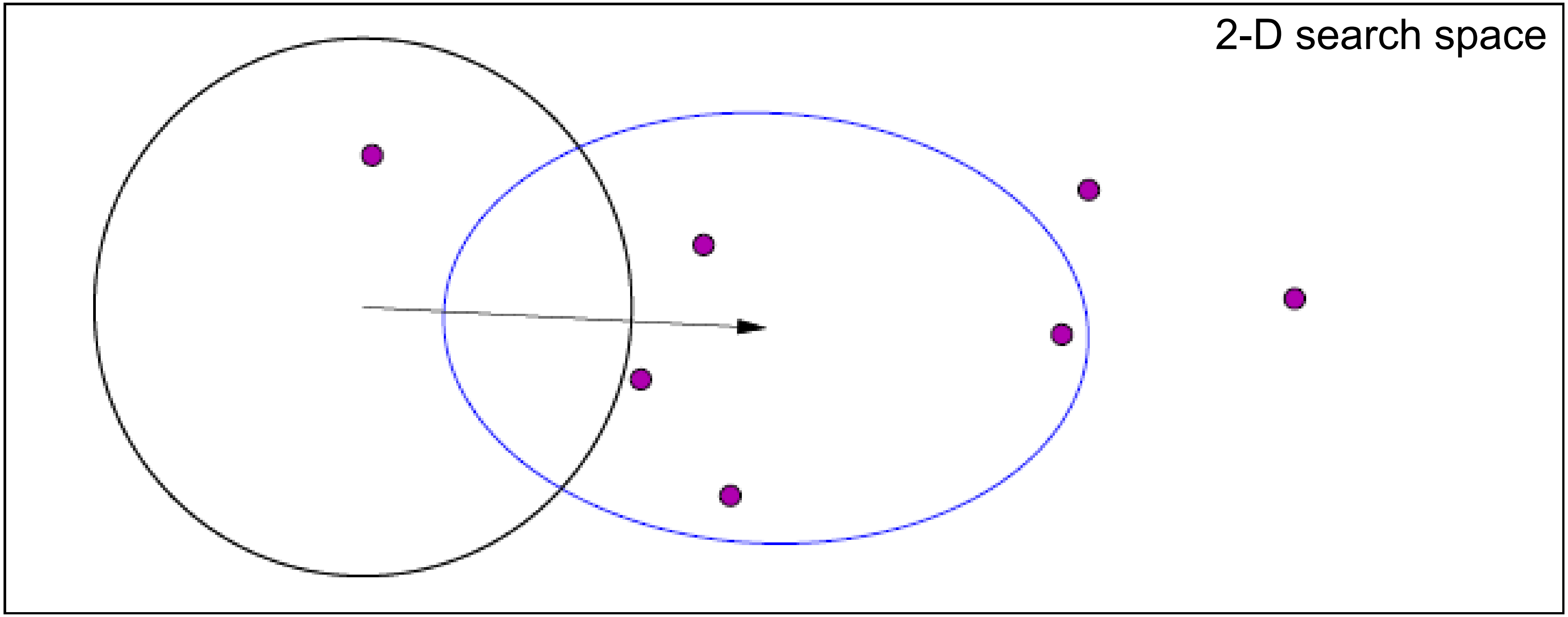


new distribution $\mathcal{N}(m, \mathbf{C})$ (disregarding σ)

.

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

Covariance Matrix Adaptation

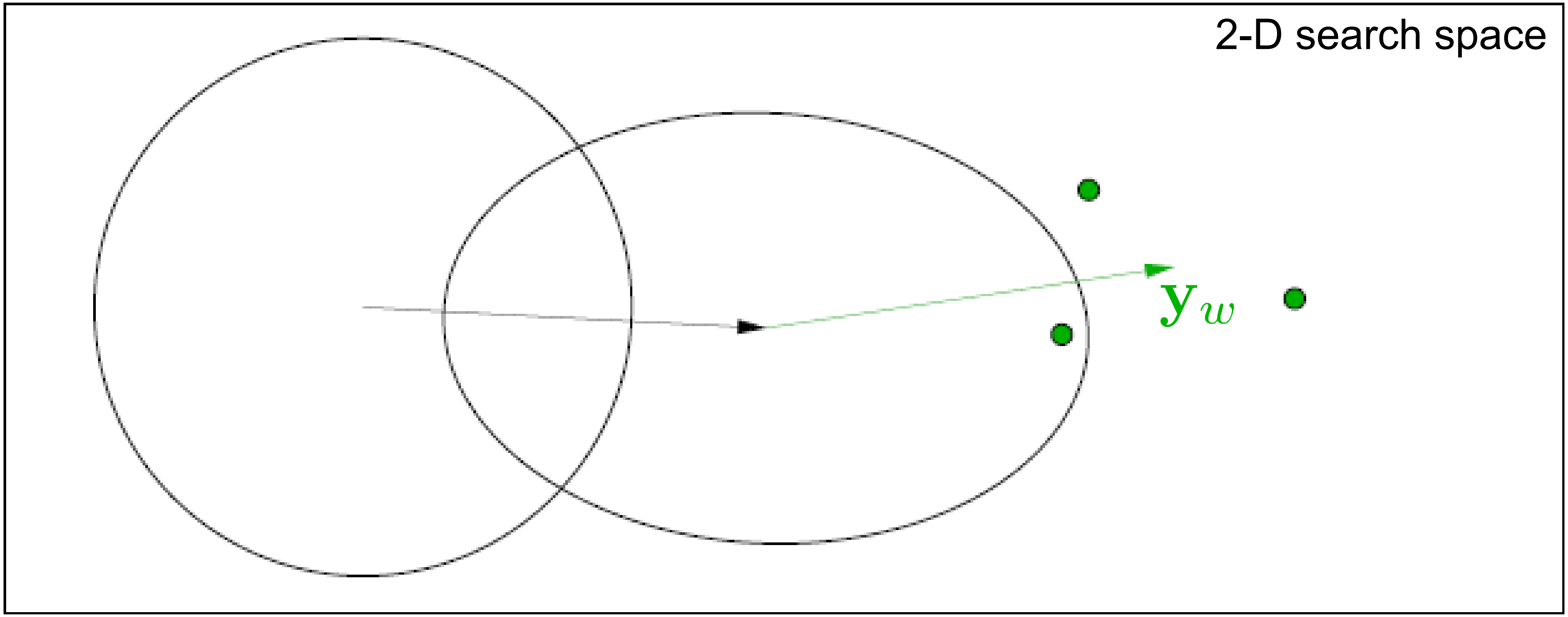


another sample $\mathbf{x}_i \sim \mathcal{N}_i(m, \mathbf{C})$

.

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

Covariance Matrix Adaptation

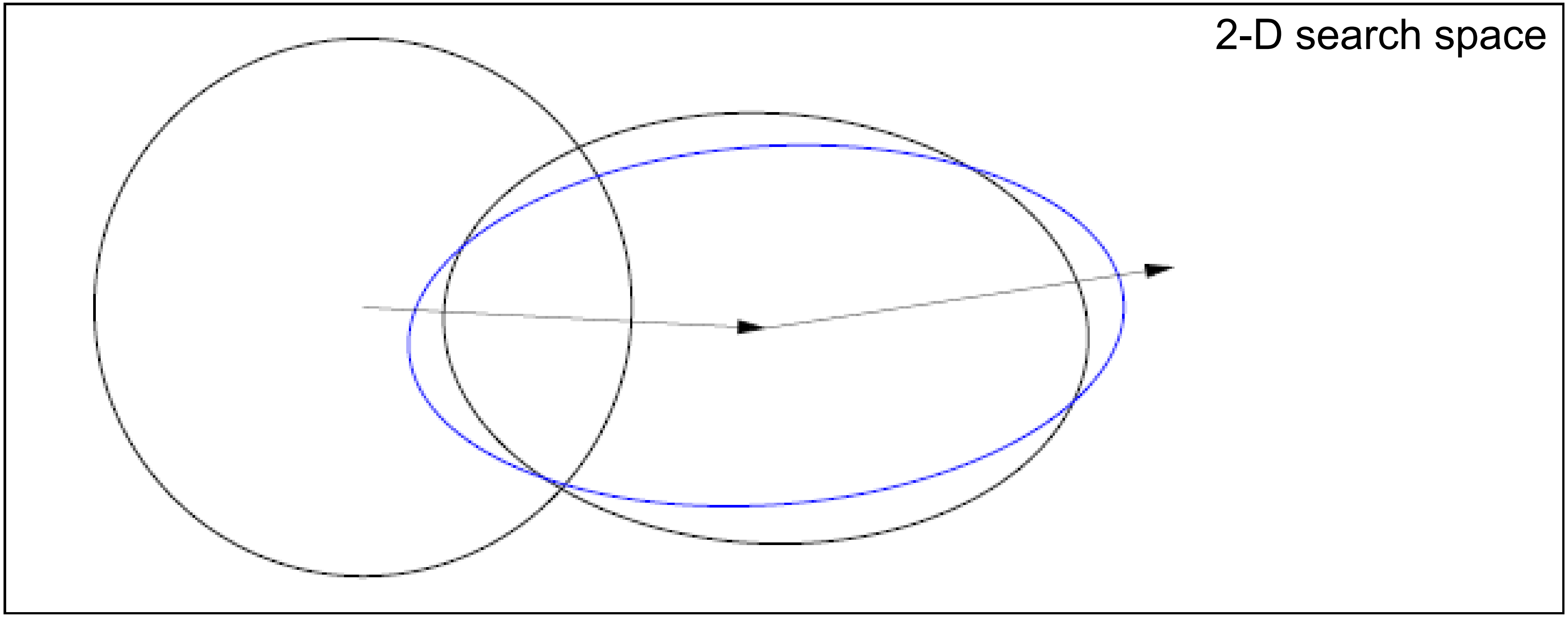


$\mathbf{y}_w$, movement of the population mean m

.

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

Covariance Matrix Adaptation



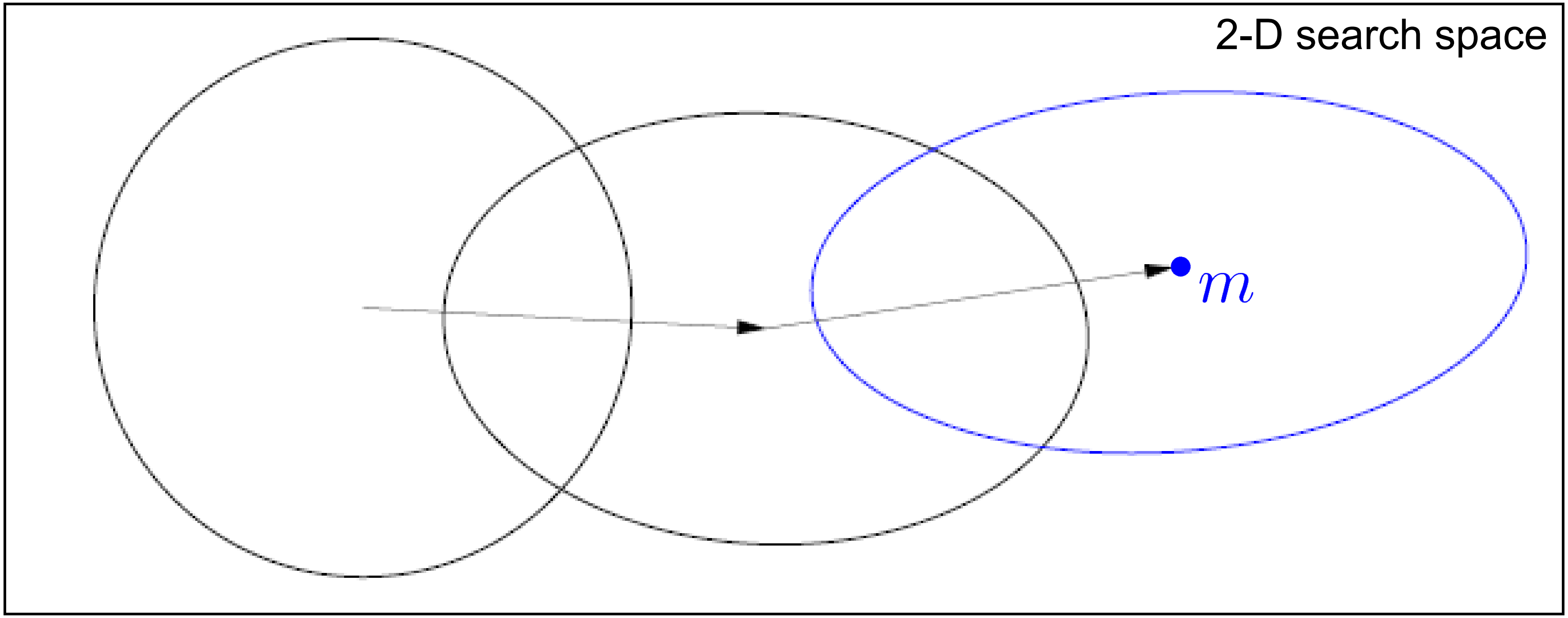
mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

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

.

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

Covariance Matrix Adaptation



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

ruling principles:

- increase the likelihood
  - of successful points by updating $m \leftarrow m + \mathbf{y}_w$
  - of successful steps by updating $\mathbf{C} \leftarrow 0.8\,\mathbf{C} + 0.2\,\mathbf{y}_w\mathbf{y}_w^{\mathrm{T}}$
- increase $\mathbb{E}[w_k(f(x))]$ by natural gradient descent in m and $\mathbf{C}$

[Kjellstroem 1991, Hansen&Ostermeier 1996, Ljung 1999]

Interpretations/Observations

- learning pairwise dependencies between all variables
- natural gradient ascent on $\theta \iff$ right metric in p_θ-space

$$\theta^{k+1} = \theta^k + \eta \widetilde{\nabla}_\theta \widehat{\mathbb{E}}(w(f(x))) \qquad x \sim P(.|\theta^k)$$

- conducting a principle component analysis (PCA) of steps, sequentially in time and space

  eigenvectors of the covariance matrix are the principle components that are sampled independently

- adaptive representation, adaptive encoding in x-space

  principle components define a new coordinate system

- resembles quasi-Newton methods, variable metric in x-space

  the covariance matrix defines a Mahalanobis metric, it adapts to the inverse Hessian of f

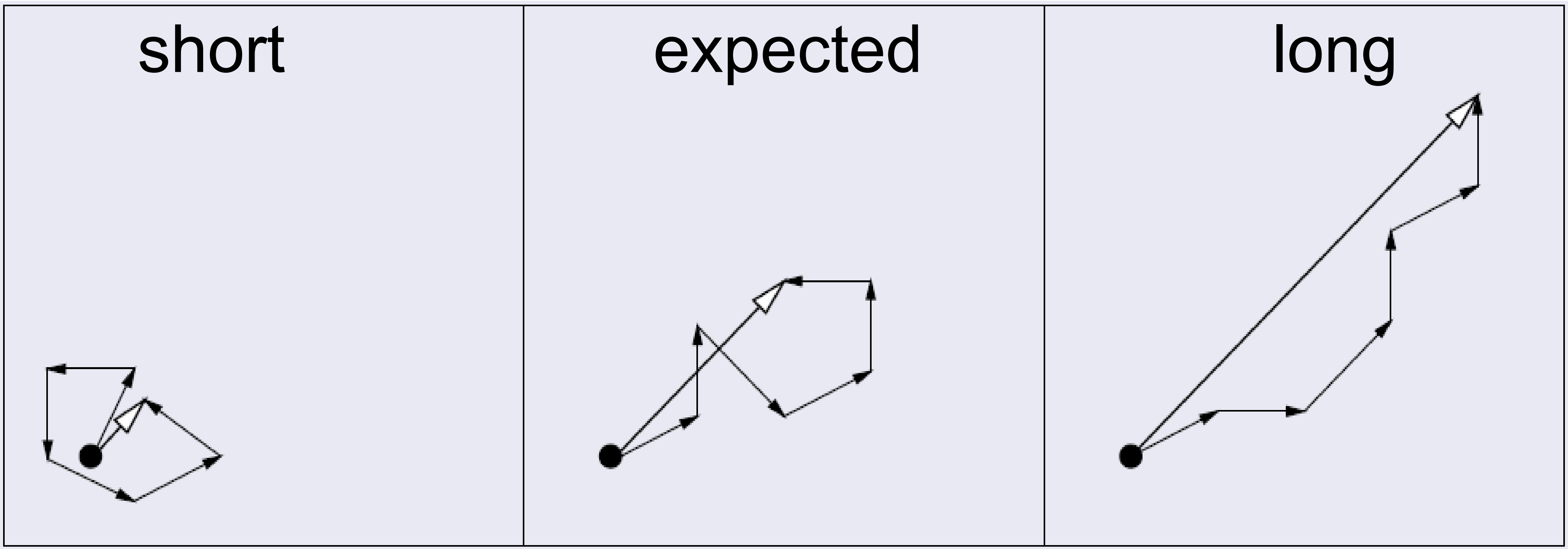
- entirely independent of the coordinate system

  algebraic formulation, invariance is a major design principle

Step-size control: the concept

search paths in 2-D

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

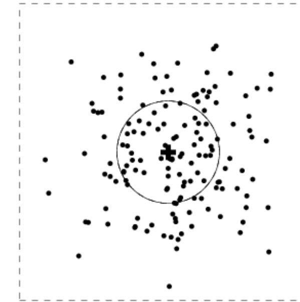


too large step-size | neutral and optimal step-size | too small step-size

if several updates go into the same/similar direction (if they have the same sign) the step-size is increased

CMA-ES in a nutshell

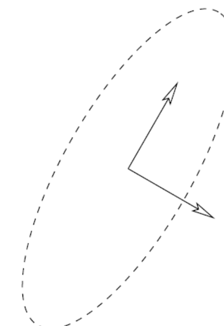
1) Sample maximum entropy distribution



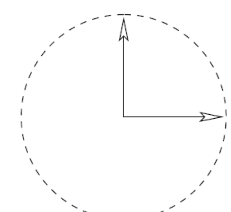
$$\mathbf{x}_i = m + \sigma \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$ multivariate normal distribution

2) Ranking solutions according to their fitness

invariance to order-preserving transformations



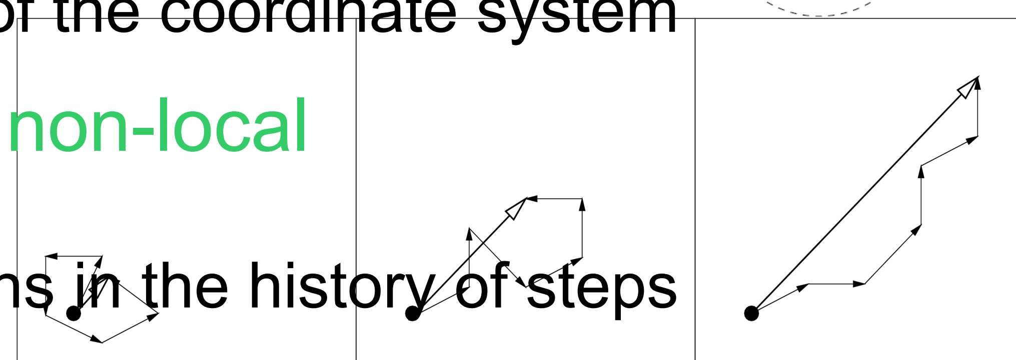
3) Update mean and covariance matrix by increasing the likelihood of good points and steps (also: natural gradient ascend)



PCA → variable metric,
new problem representation,
invariant under changes of the coordinate system

4) Update step-size based on non-local information

exploit correlations in the history of steps



CMA-ES (Covariance Matrix Adaptation Evolution Strategy)
= natural gradient ascent + cumulation + step-size control

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\lambda \in \{2, 3, 4, \dots\}$, usually $\lambda \geq 5$

Set $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1$,
set $w_{i=1,\dots,\lambda}$ decreasing in i, $\sum_i |w_i| = 1$ and $\mu_w^{-1} := \sum_i {w_i}^2 \approx 3/\lambda$
Initialize $\mathbf{C} = \mathbf{I}$, and $\mathbf{p}_c = \mathbf{0}$, $\mathbf{p}_\sigma = \mathbf{0}$

While not *terminate*

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

$$m \leftarrow m + \sigma \sum_i w_{\rho(i)}\, \mathbf{y}_i =: m + \sigma\, \mathbf{y}_w, \qquad \text{update mean}$$

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

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

$$\mathbf{p}_c \leftarrow (1 - c_c)\, \mathbf{p}_c + \mathbb{1}_{[0,2n]}\left\{\|\mathbf{p}_\sigma\|^2\right\} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, \mathbf{y}_w \qquad \text{path for } \mathbf{C}$$

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

[Hansen&Ostermeier 2001, Hansen et al 2003, Hansen&Kern 2004]

Design principles applied for CMA-ES

- Minimal prior assumptions

  stochastic helps, maximum entropy distribution
  improvement only by selection of solutions
  $\Longrightarrow$ harder to deceive

- Exploit all available information

  given remaining design principles (e.g. invariance)
  e.g. cumulation exploits "sign" information

- Stationarity or unbiasedness

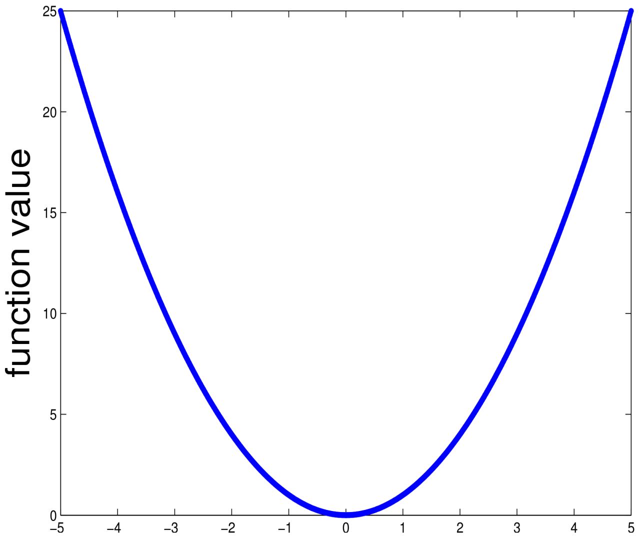
  parameters remain unchanged under "random" ranking

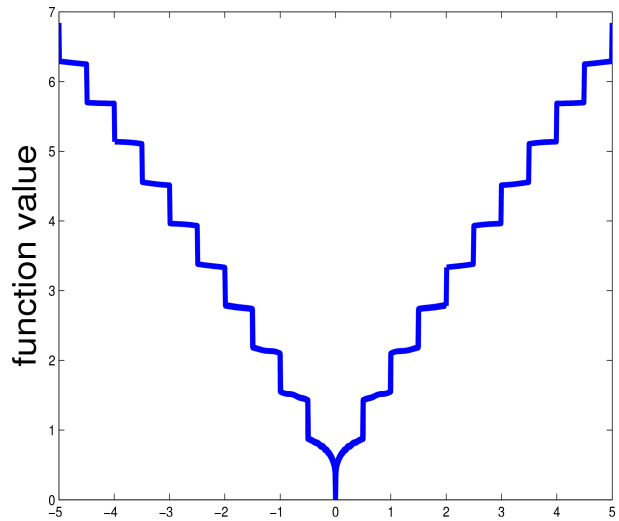
- Almost parameter-less

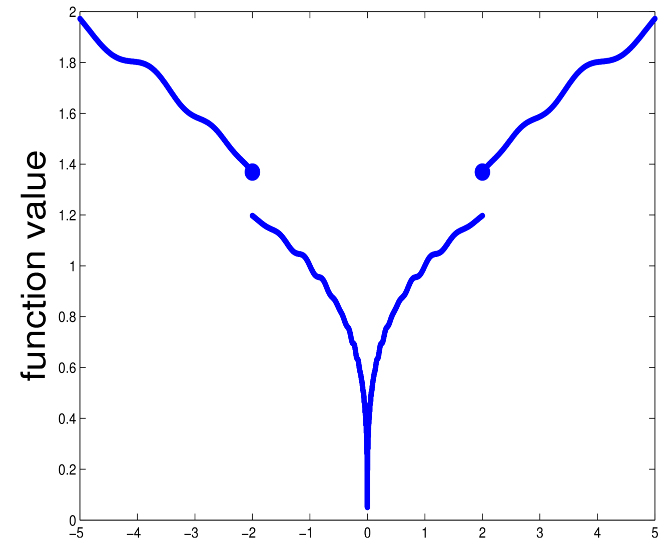
  meaningful parameters whose choice is f-independent,
  e.g. learning rates (time horizons)

- Retain and introduce invariance properties

Invariance principle: an example







Three functions belonging to the same equivalence class

Theorem (Invariance to order preserving transformation). *Given the objective function $f = g \circ h$, CMA-ES is invariant under the choice of a strictly increasing g. The set of functions $\{f : \mathbb{R}^n \to \mathbb{R} \mid f = g \circ h$ with g strictly increasing$\}$ is an equivalence class with indistinguishable search trace.*

→ derivative *and function-value* free

...experimental validation...

CMA-ES is regarded as “state-of-the-art”

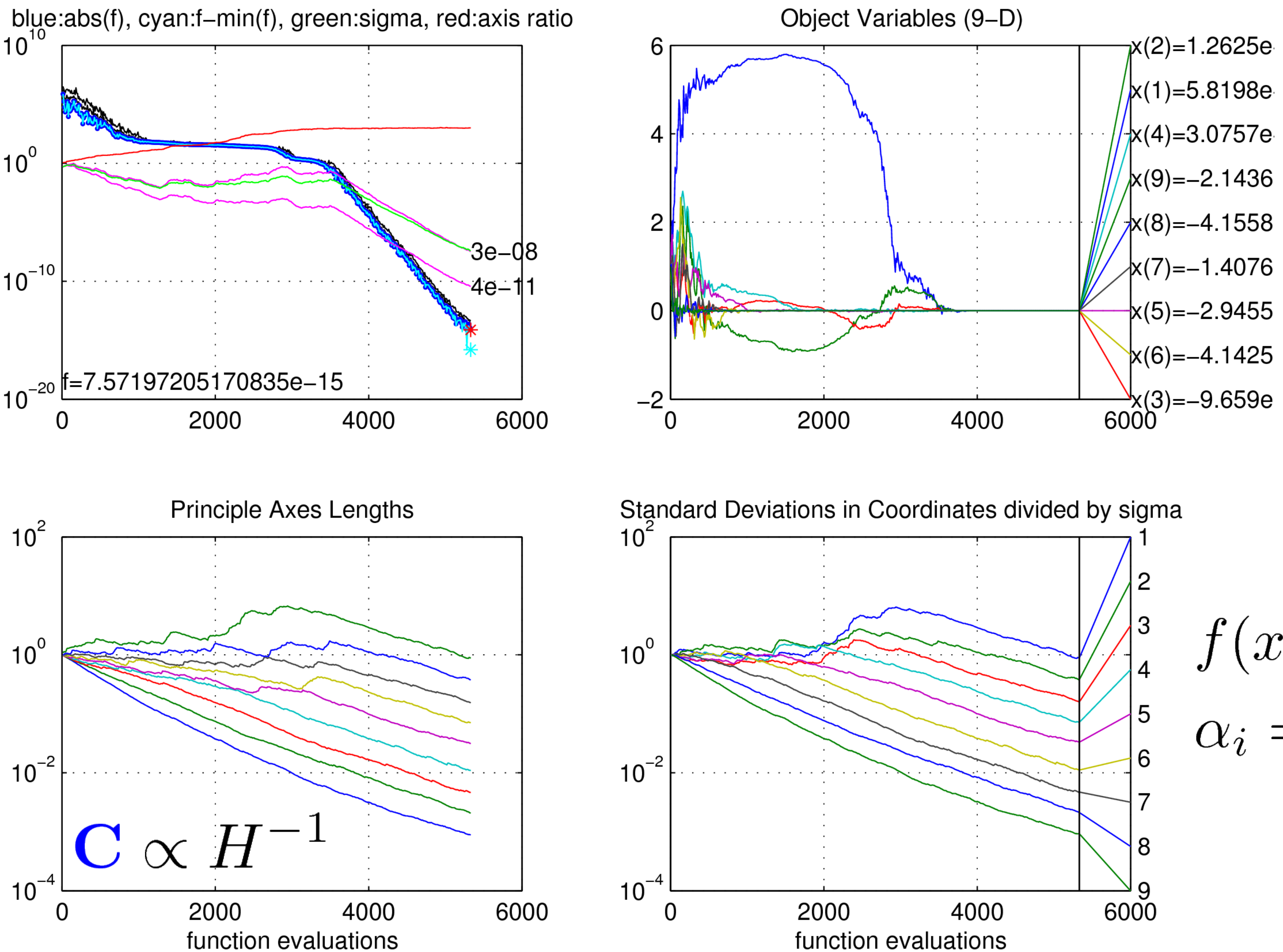
- > 1000 citations to the two seminal papers
- $\gg 100$ applications published
- implemented in libraries for
  - evolutionary computation [EO, Beagle,. . .]
  - pattern search [NOMADm]
  - machine learning [Shark]
  - robotics [PACLib]
  - chart analysis [AmiBroker]
  - water model calibration [PEST]
- > 20 daily hits to the source code download page

A simple unimodal test function

$$f(x) = g\left(\frac{1}{2}x^T H x\right)$$

- for different strictly monotonic (i.e. order-preserving) $g : \mathbb{R} \to \mathbb{R}$
- with uniform eigenspectrum of the Hessian H
- with different condition numbers of H (ratio between largest and smallest eigenvalue) between one and 10^{10}
- in dimension 9 and 20

Experimentum crucis

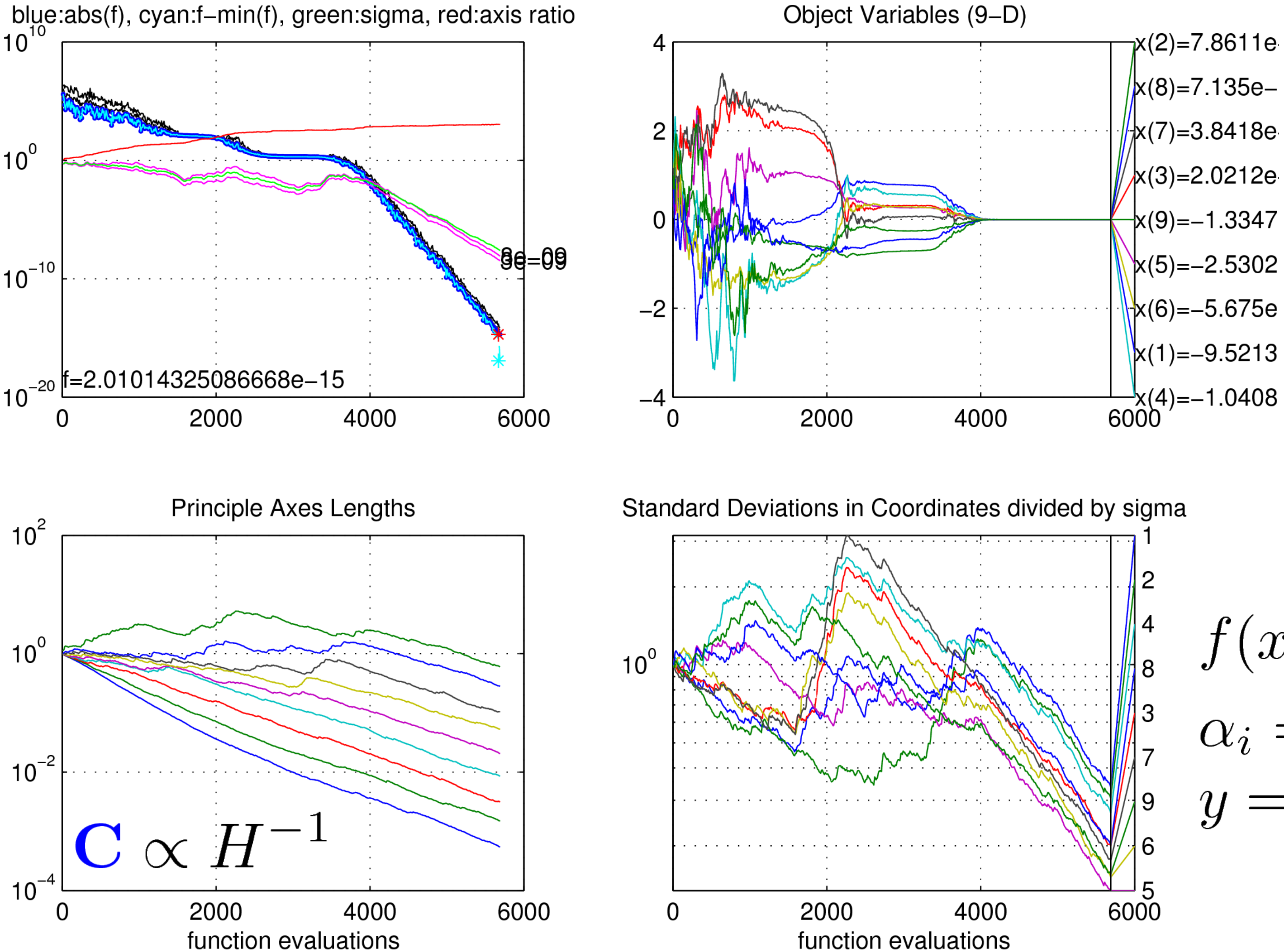


$$f(x) = \sum_{i=1}^{n} \alpha_i x_i^2$$

$$\alpha_i = 10^{6\frac{i-1}{n-1}}$$

without covariance matrix adaptation it takes 1000 times longer to reach $f = 10^{-10}$

Experimentum crucis



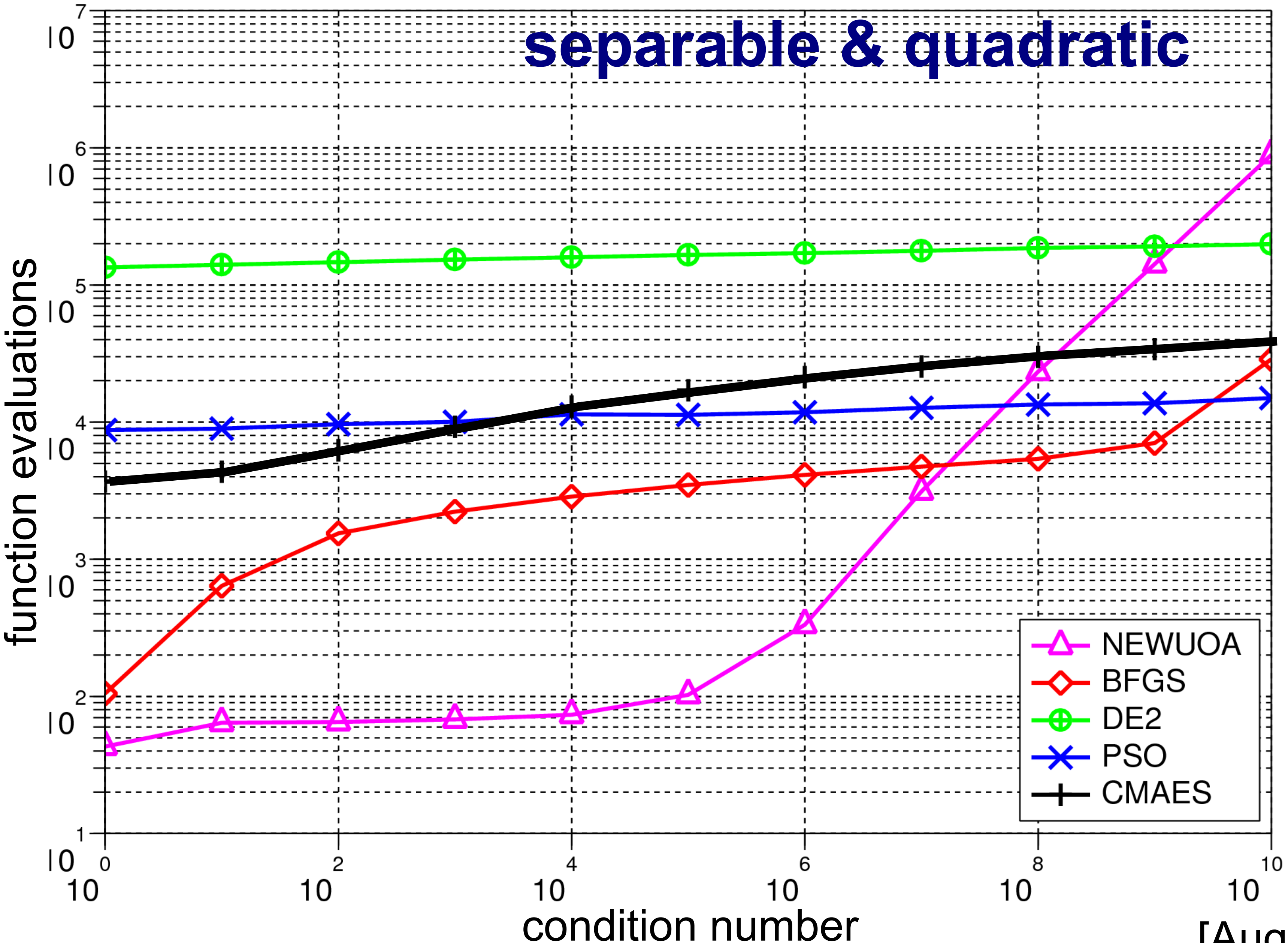
$$f(x) = \sum_{i=1}^{n} \alpha_i y_i^2$$

$$\alpha_i = 10^{6\frac{i-1}{n-1}}$$

$$y = \mathrm{rotation}(x)$$

without covariance matrix adaptation it takes 1000 times longer to reach $f = 10^{-10}$

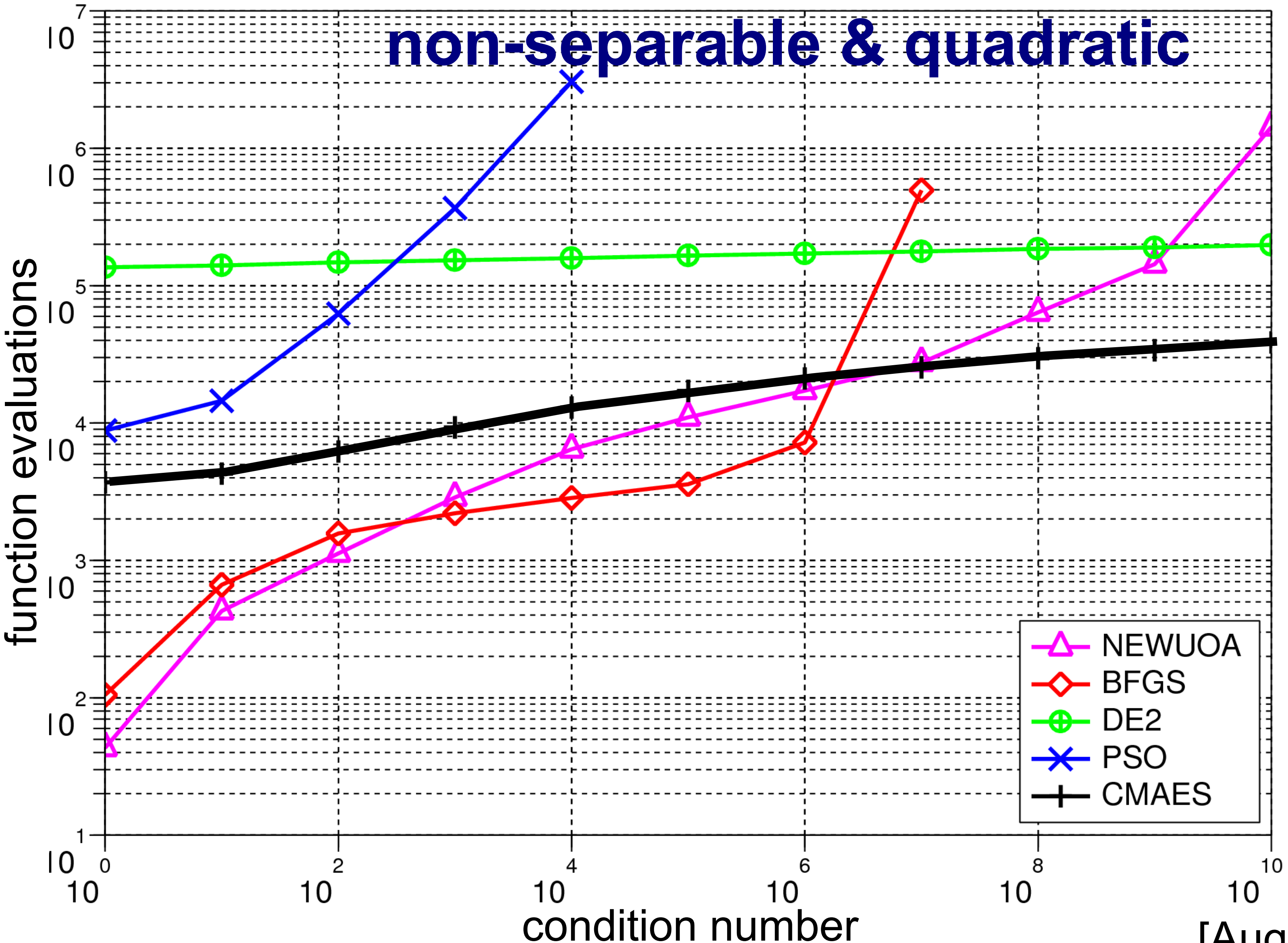
Runtime versus condition number



1

[Auger et al 2009]

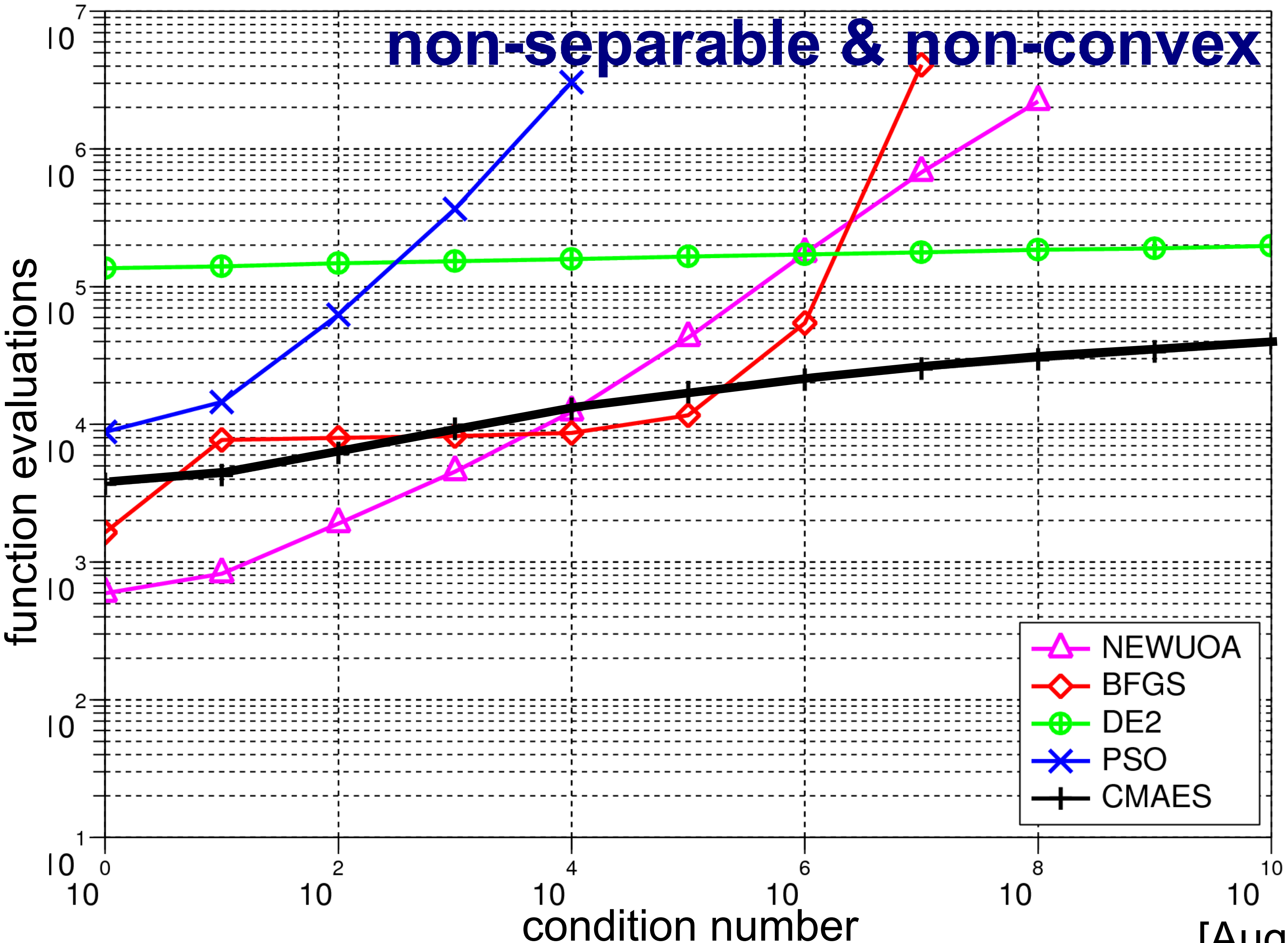
Runtime versus condition number



2

[Auger et al 2009]

Runtime versus condition number



3

[Auger et al 2009]

COCO/BBOB test environment

COCO: COmparing Continuous Optimizers

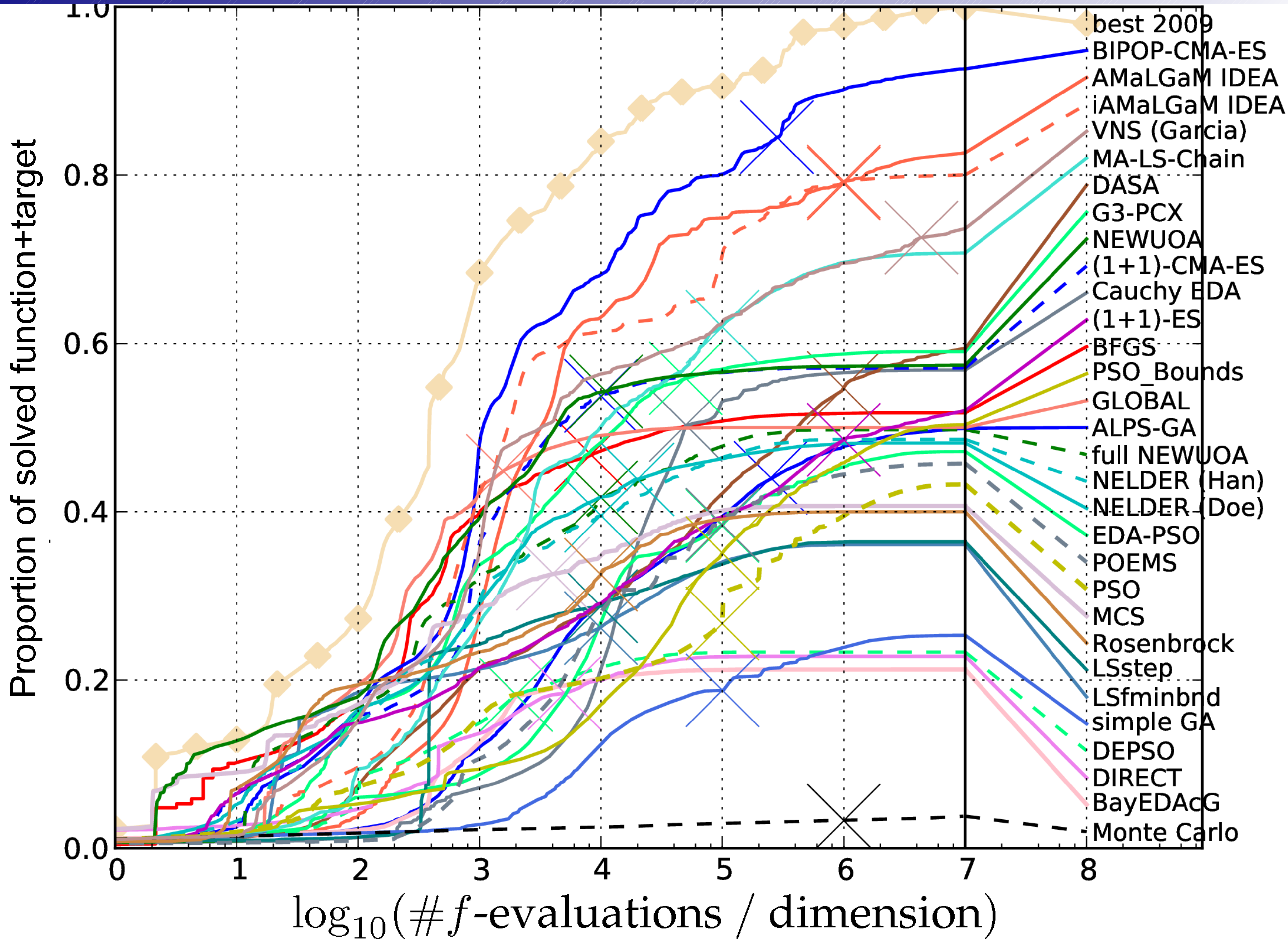
BBOB: Black-Box Optimization Benchmarking

- 24 test functions with a wide range of difficulties
- 30 noisy functions

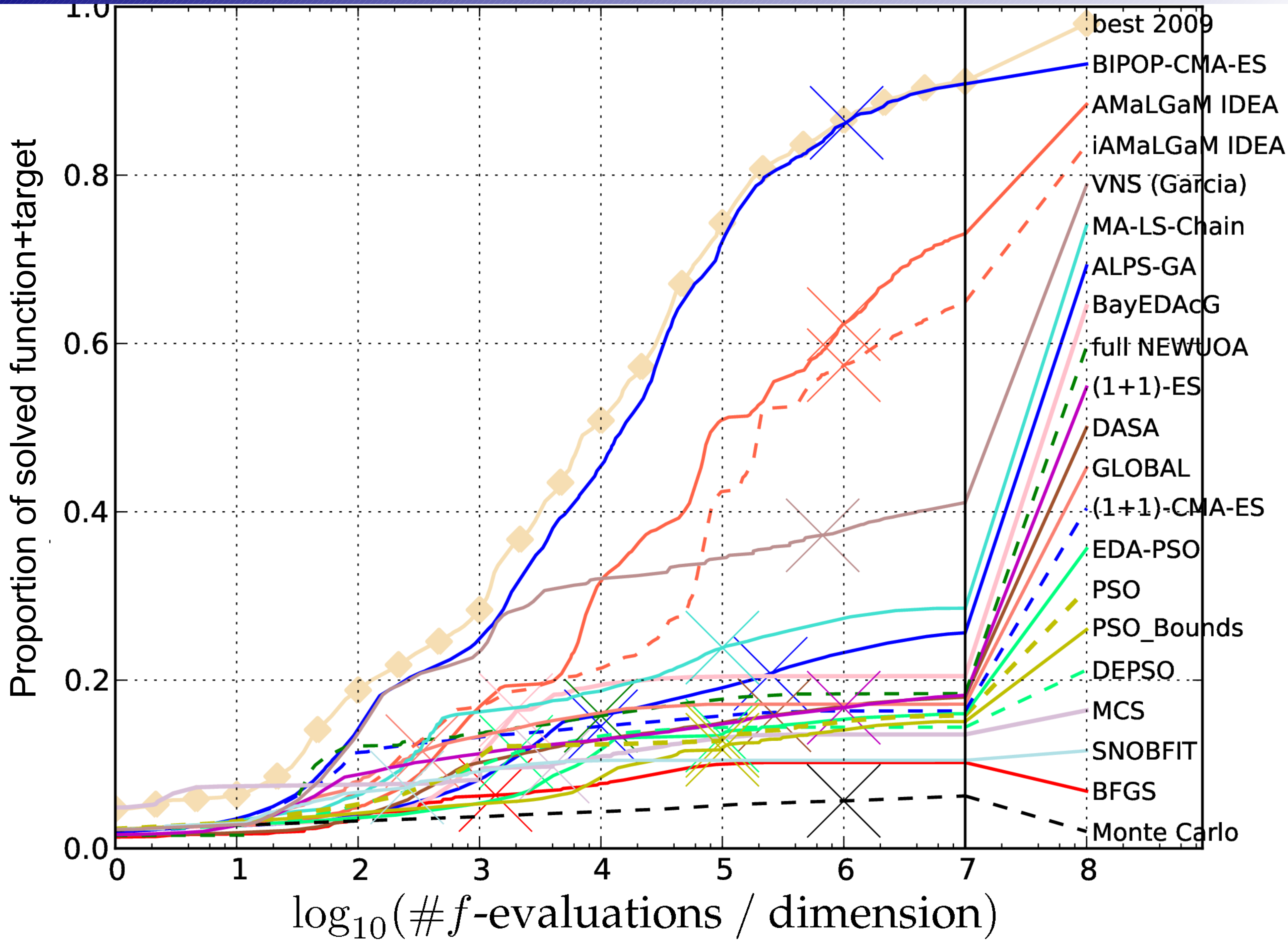
30+ algorithms have been tested:

http://coco.gforge.inria.fr

Results of BBOB-2009 (noisefree, 20-D)



Results of BBOB-2009 (noisy, f_{101}-f_{130}, 20-D)



Limitations of CMA-ES

- internal CPU-time: $10^{-8}n^2$ seconds per function evaluation on a 2GHz PC, tweaks are available

  1 000 000 f-evaluations in 100-D take 100 seconds *internal* CPU-time

- better methods are presumably available in case of
  - partly separable problems
  - specific problems, for example with cheap gradients

    specific methods
  - small dimension ($n \ll 10$)

    for example Nelder-Mead
  - small running times (number of f-evaluations $\ll 100n$)

    model-based methods

Questions?

CMA-ES source code: http://www.lri.fr/~hansen/cmaes_inmatlab.html