The CMA-ES (and its application to space flight trajectory optimization)

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

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

...please 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

The Problem: Black-Box Optimization

Black-Box Optimization (Search)

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

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

in a black-box scenario (direct search)

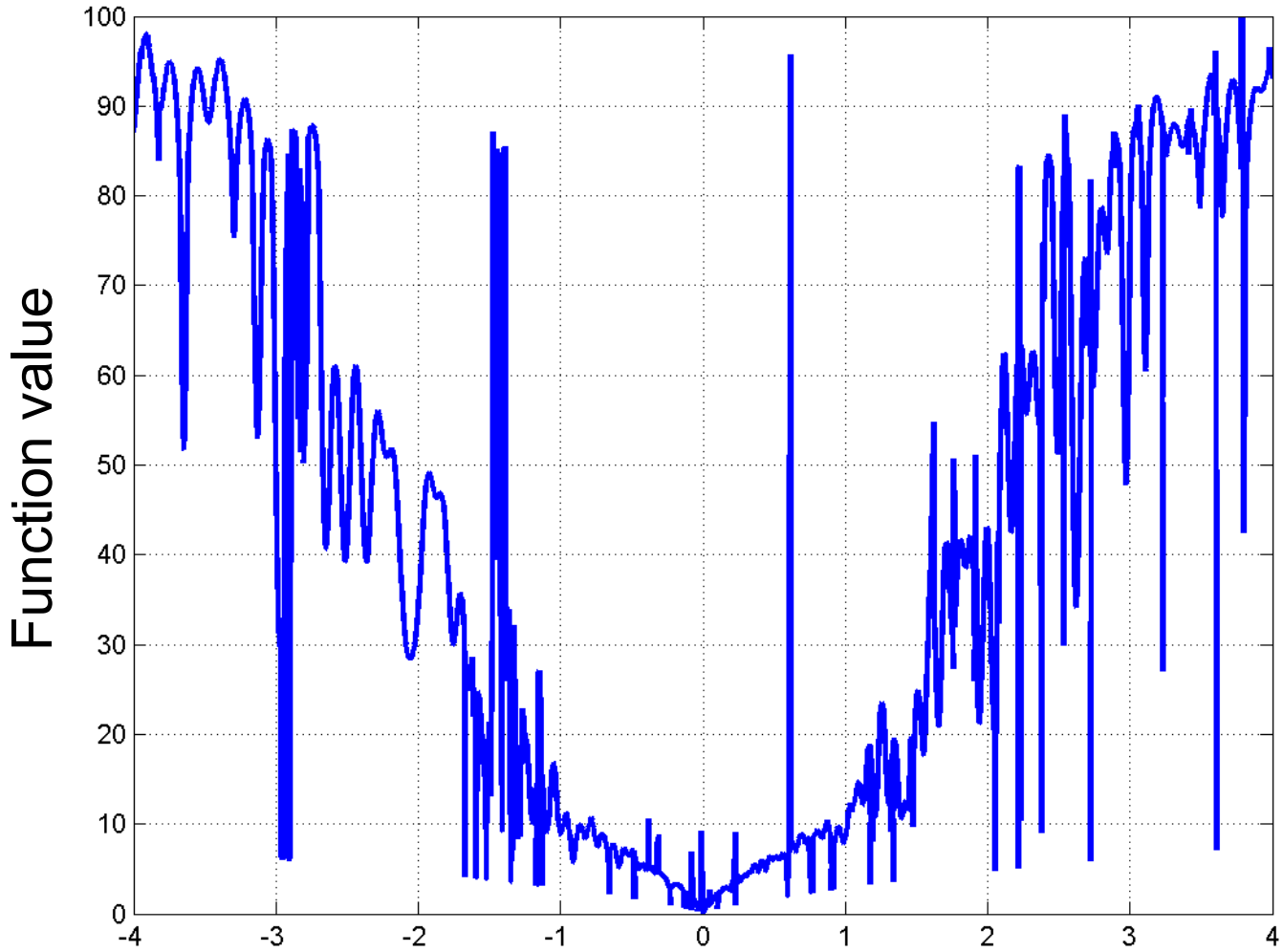
where

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

The search costs are the number of back-box calls (function evaluations)

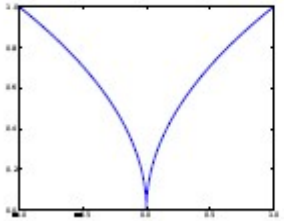
Rugged landscape

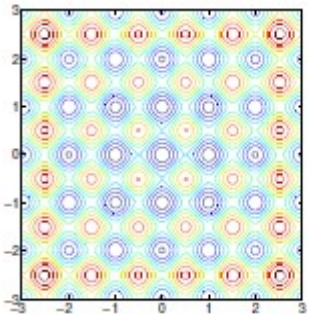
Section through 5-D ($n = 5$) landscape

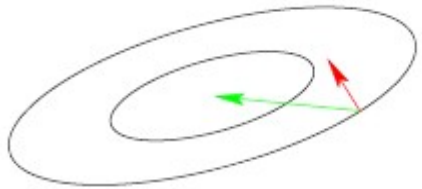


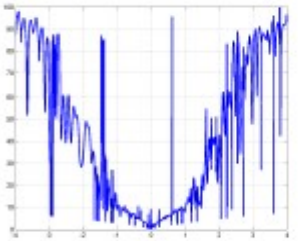
Difficulties in black-box optimization

- 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

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 search methods

- Evolutionary algorithms [Rechenberg 1965]
- 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
  - operate on continuous search spaces
  - with problem-tailored encoding of the fitness
- Genetic programming
  - operates on trees
  - evolving computer programs

Metaphores

(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

...a stochastic optimization method...

Stochastic optimization template

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

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

$$\theta \leftarrow Update(\theta, [x_1, \widehat{w}(f(x_1))], \ldots, [x_\lambda, \widehat{w}(f(x_\lambda))])$$

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 terminate

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

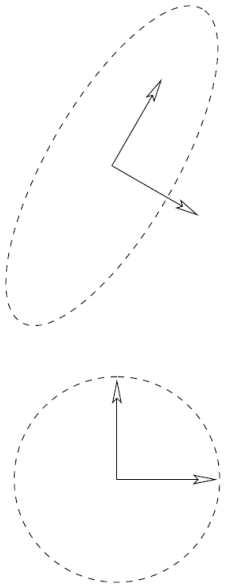
Return, e.g., the expected value of P: $m \in \theta$

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

Instantiation of the template

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

➔ CMA-ES (covariance matrix adaptation evolution strategy) [Hansen&Ostermeier 1996, 2001, Hansen et al 2003, Hansen&Kern 2004, Jastrebski&Arnold 2006]

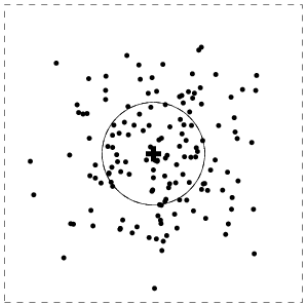
➔ Natural Evolution Strategy (theory-driven)

[Wierstra et al 2008, Sun et al 2009, Glasmachers et al 2010]

CMA-ES in a nutshell

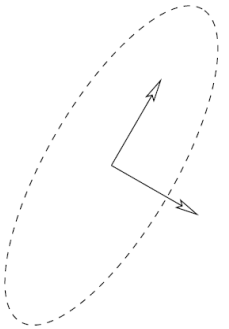
1) Sample maximum entropy distribution

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



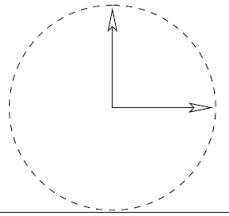
2) Ranking solutions according to their fitness

invariance to order-preserving transformations



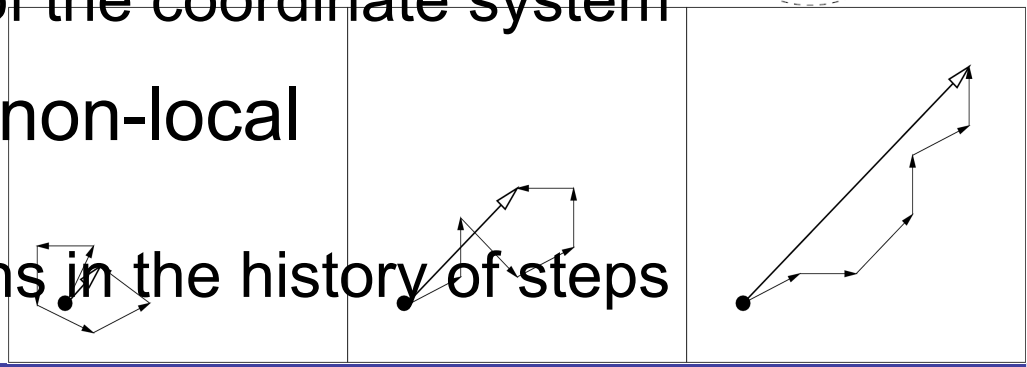
3) Update mean and covariance matrix by natural gradient ascend, improving the "expected fitness" and the likelihood for good steps

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

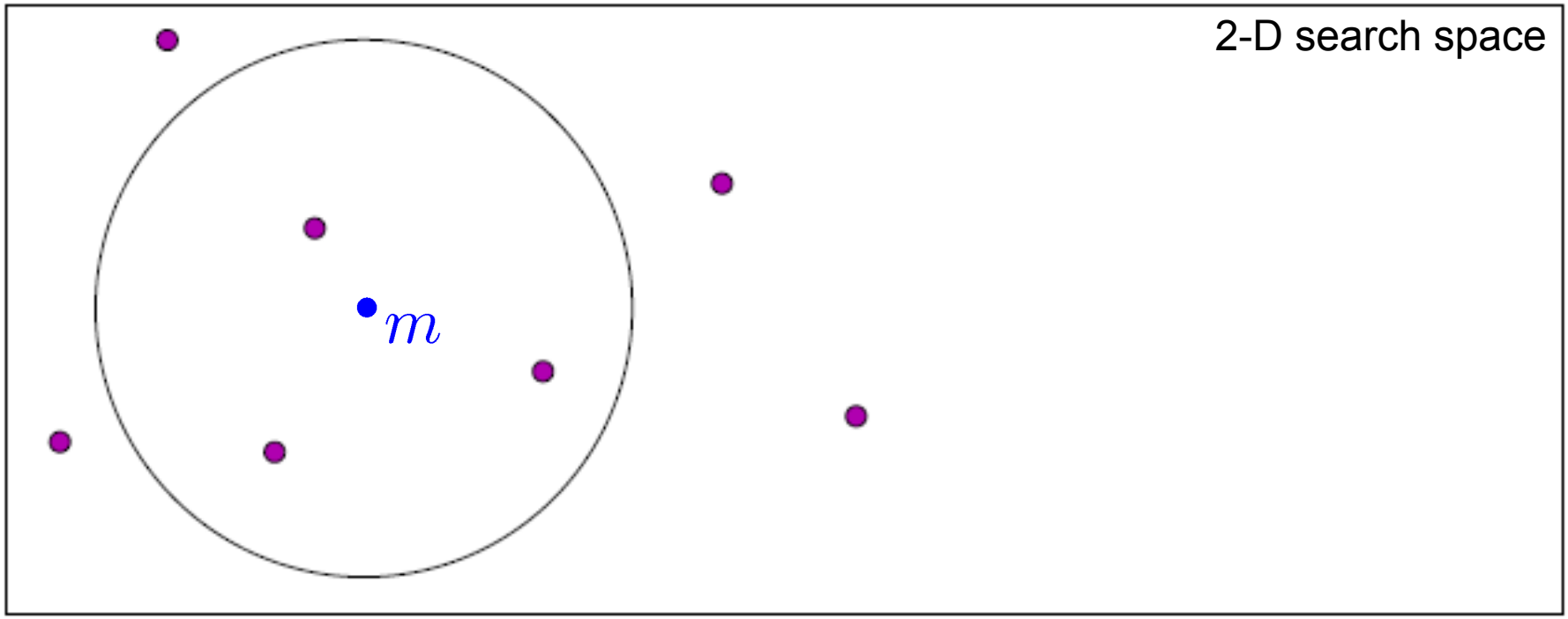


4) Update step-size based on non-local information

exploit correlations in the history of steps



Covariance Matrix Adaptation

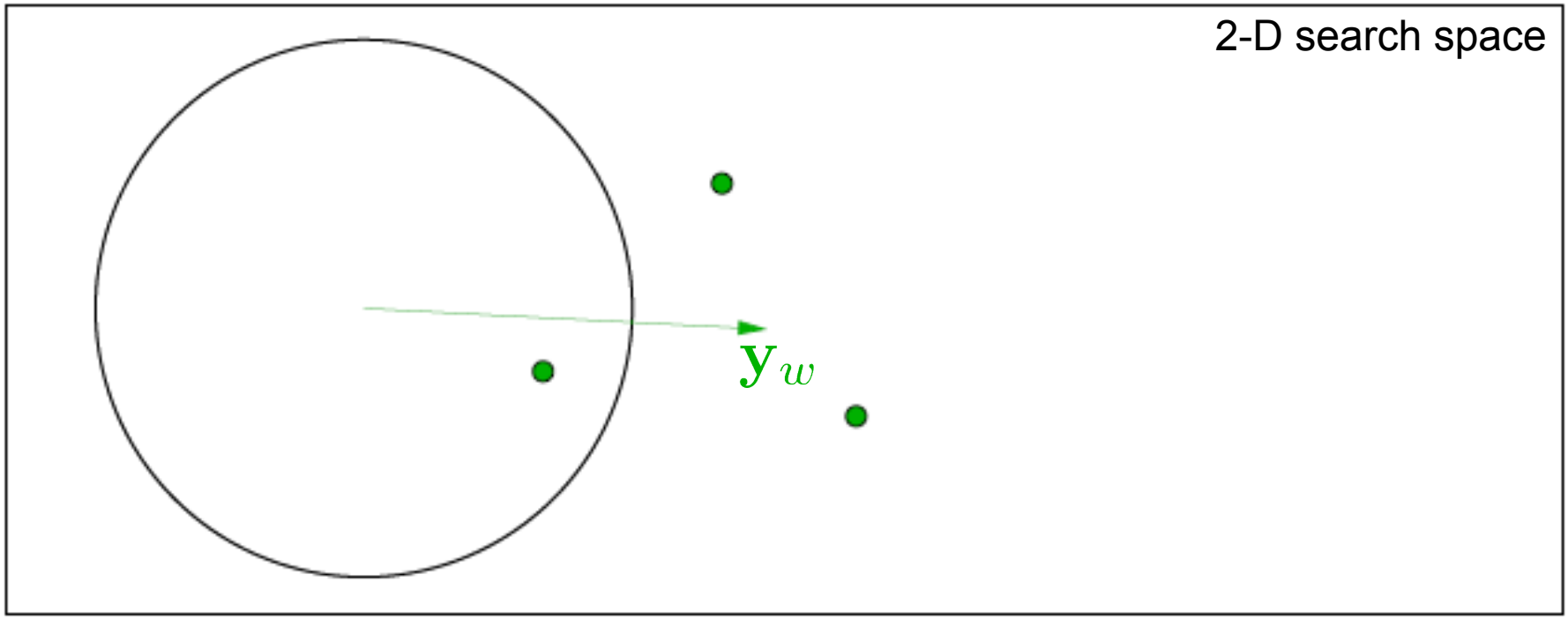


sample from the initial distribution: $\mathcal{N}(m, \mathbf{I})$, $\mathbf{C} = \mathbf{I}$

.

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

Covariance Matrix Adaptation

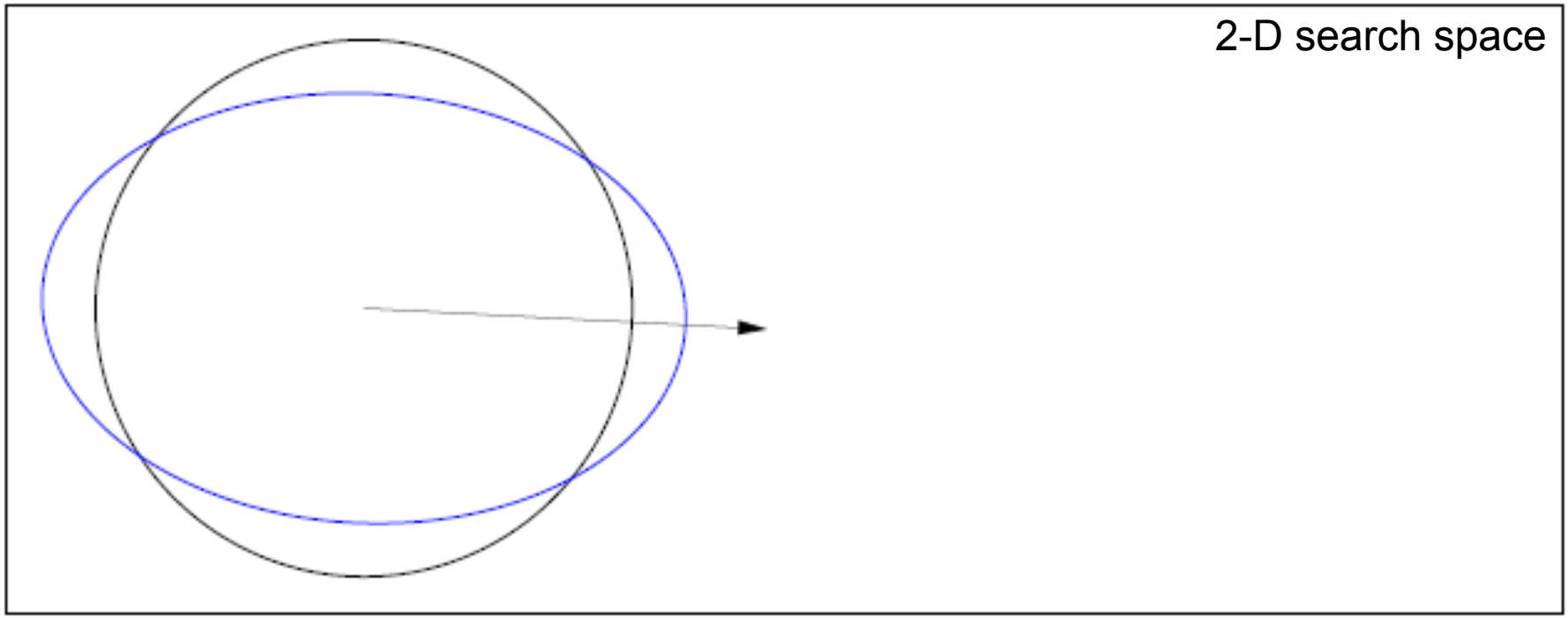


$\mathbf{y}_w$ is the move of the mean m, disregarding σ .

.

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

Covariance Matrix Adaptation



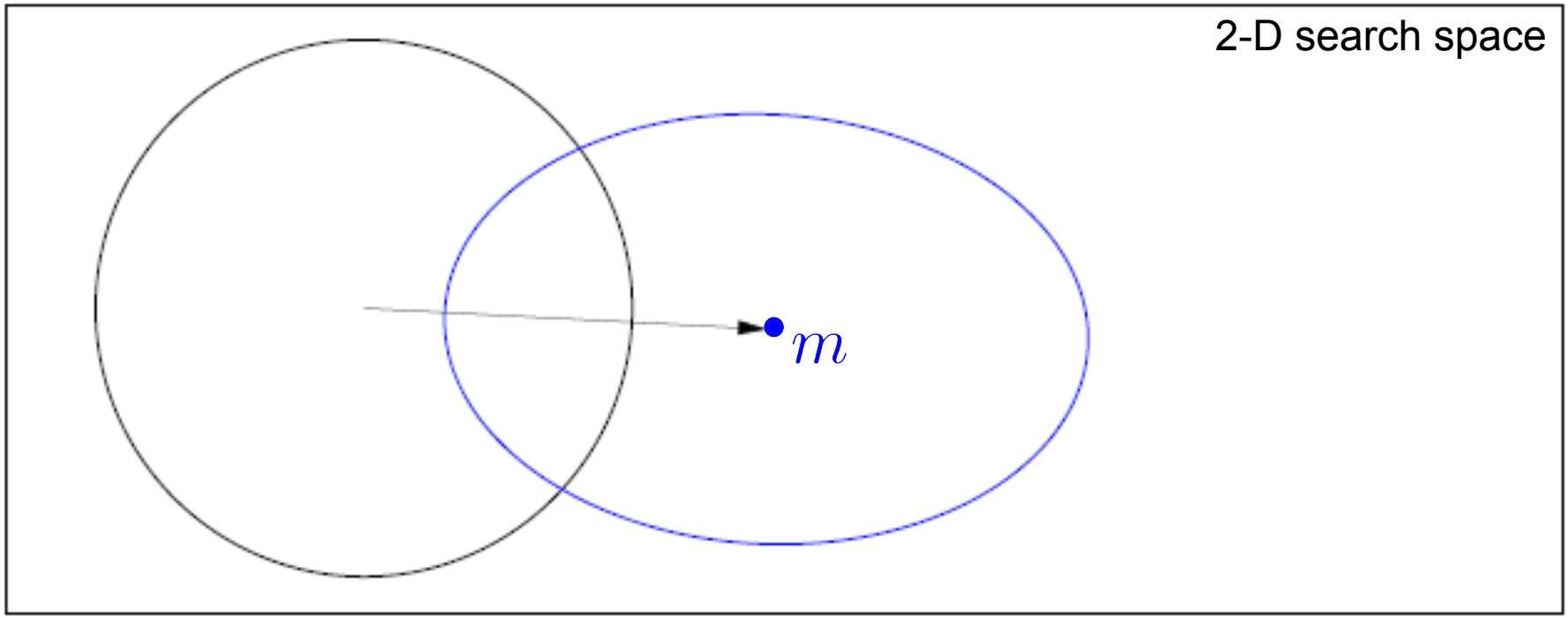
mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

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

.

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

Covariance Matrix Adaptation

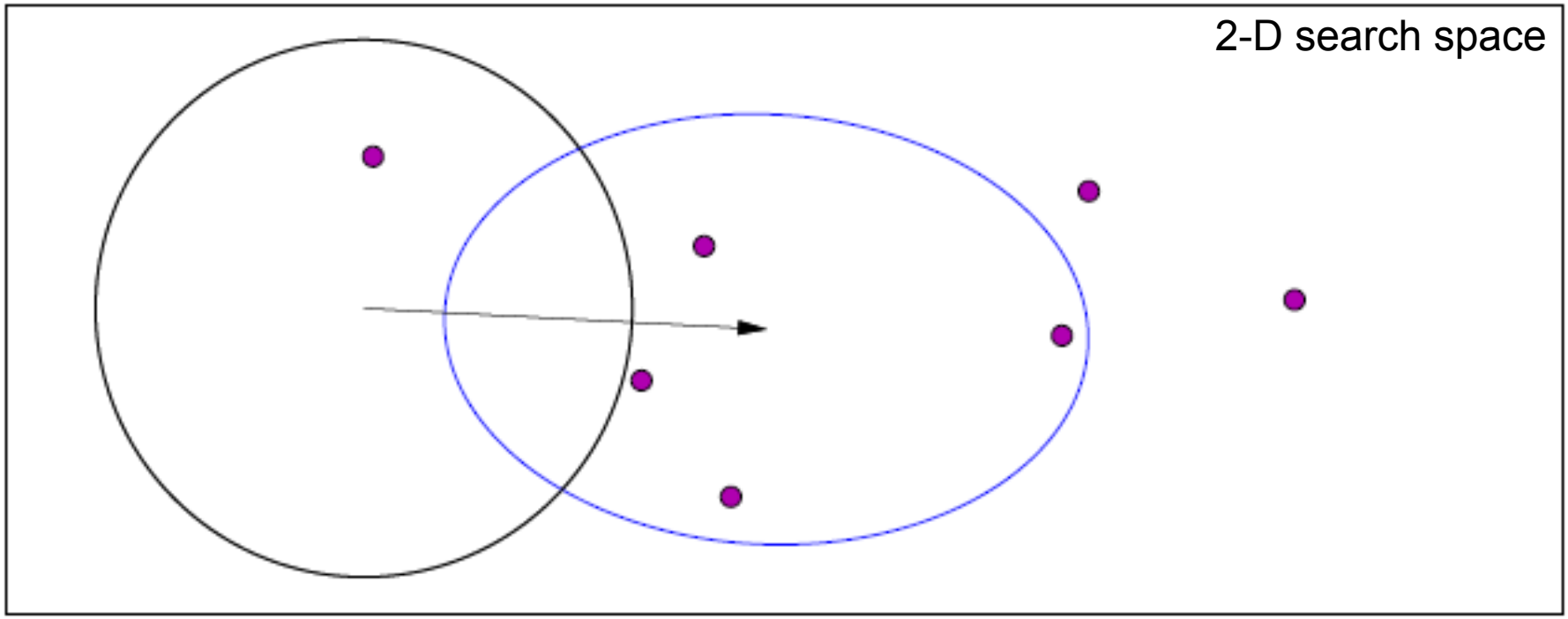


new distribution $\mathcal{N}(m, \mathbf{C})$ (disregarding σ)

.

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

Covariance Matrix Adaptation

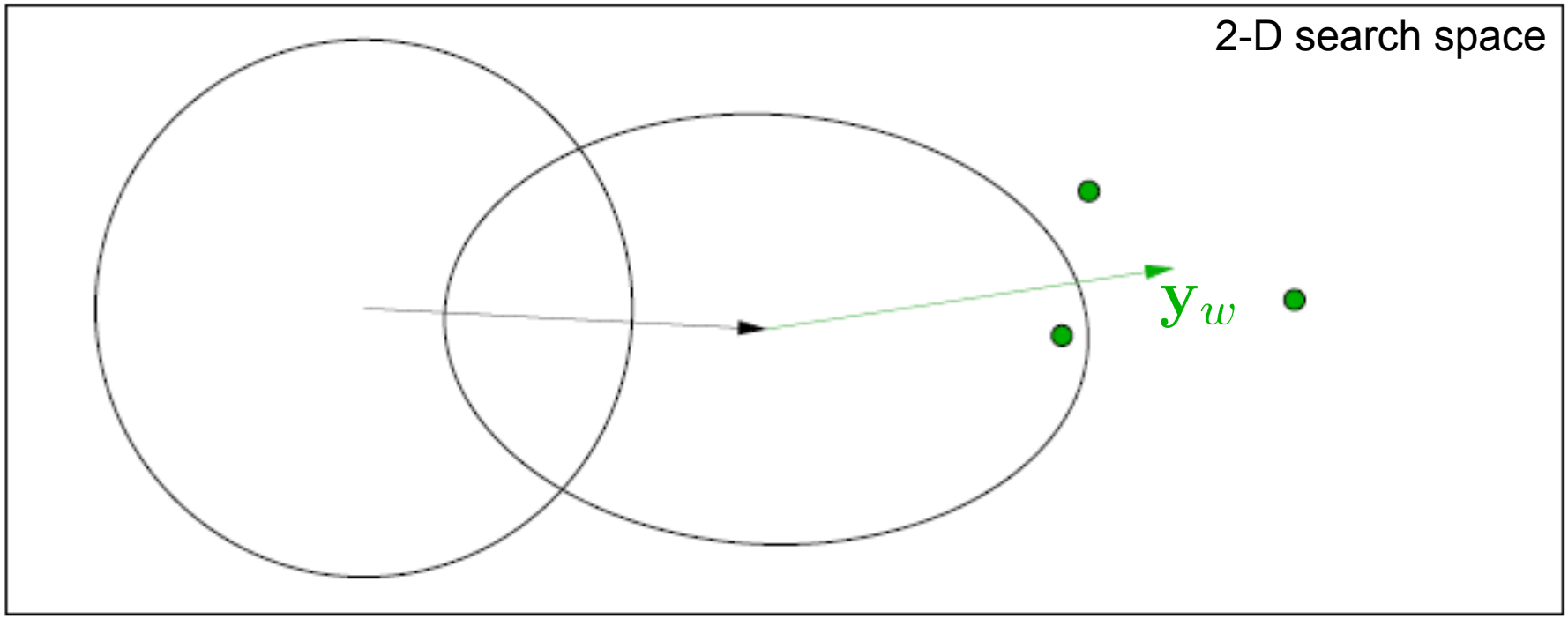


another sample $\mathbf{x}_i \sim \mathcal{N}_i(m, \mathbf{C})$

.

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

Covariance Matrix Adaptation

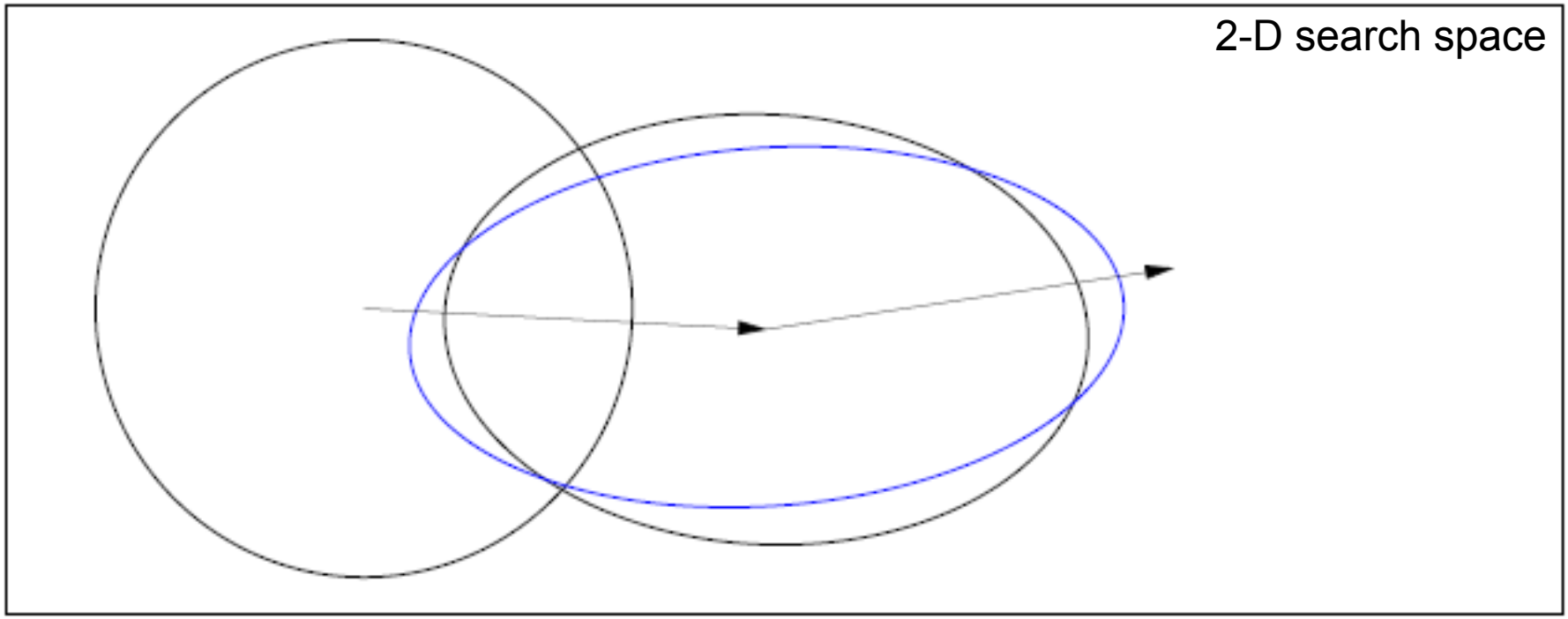


$\mathbf{y}_w$, movement of the population mean m

.

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

Covariance Matrix Adaptation



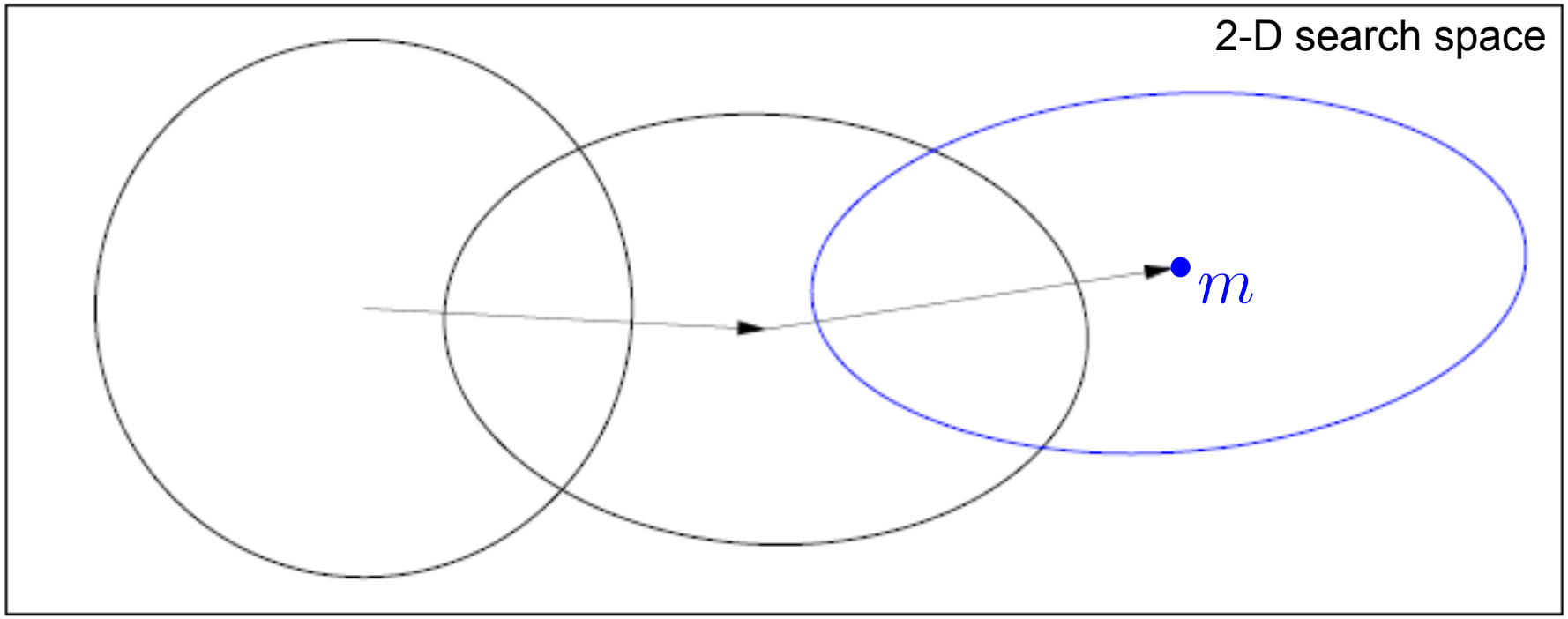
mixture of covariance matrix $\mathbf{C}$ and step $\mathbf{y}_w$

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

.

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

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

- natural gradient ascent, using the right metric in p_θ-space
- 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
- learning pairwise dependencies between all variables

  a countermeasure to epistasis
- resembles quasi-Newton methods, variable metric in x-space

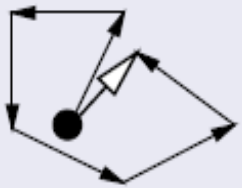
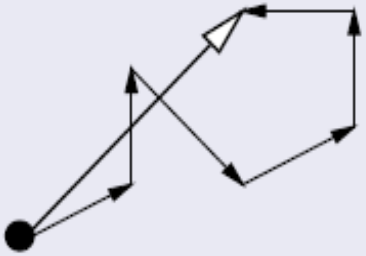
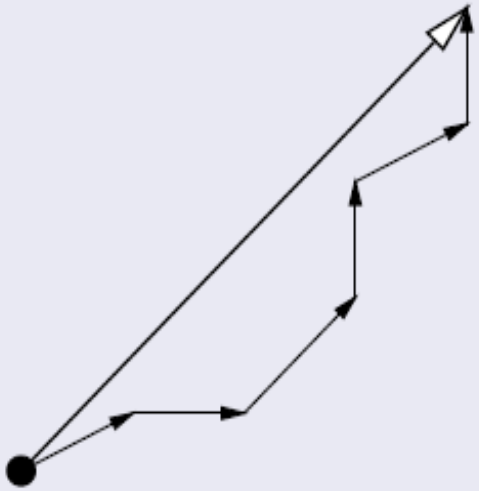
  the covariance matrix defines a Mahalanobis metric, it adapts to the inverse Hessian of f
- adaptive representation, adaptive encoding in x-space
- entirely independent of the coordinate system

  algebraic formulation, invariance is a major design principle

Step-size control: the concept

search paths in 2-D

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

short	expected	long
		
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) increase the step-size

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

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\lambda \in \{2, 3, 4, \ldots\}$, 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 $w_{i=1,\ldots,\lambda}$ 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*

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

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

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

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

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

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

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

CMA-ES: the algorithm

While not *terminate*

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

$$m \leftarrow m + \sum_{i=1}^{\lambda} \widehat{w}_{\rho(i)} \underbrace{(\mathbf{x}_i - m)}_{=\sigma\mathbf{y}_i} = m + \underbrace{\sigma\mathbf{y}_w}_{\text{iterate displacement}} \qquad \text{update mean}$$

$$\mathbf{p}_\sigma \leftarrow \underbrace{(1 - c_\sigma)}_{\text{discount factor}}\,\mathbf{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}\,\sqrt{\mu_w}\,\mathbf{C}^{-\frac{1}{2}}\,\mathbf{y}_w}_{\text{under neutral selection } \mathcal{N}(\mathbf{0},\mathbf{I})} \qquad \text{path for } \sigma$$

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

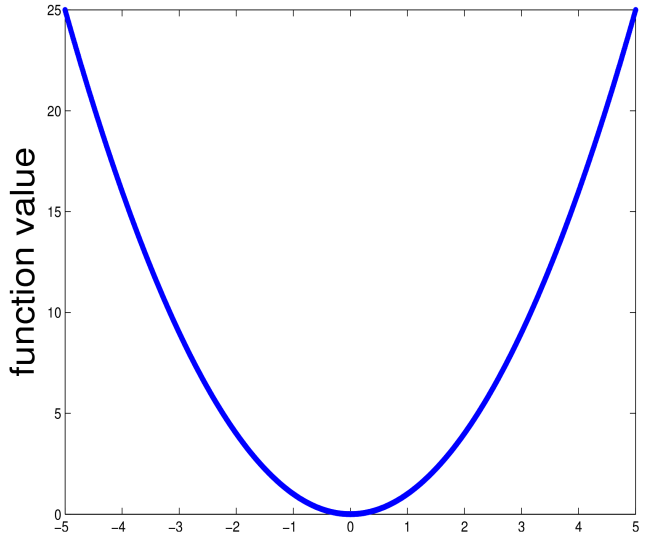
$$\mathbf{p}_\mathrm{c} \leftarrow \underbrace{(1 - c_\mathrm{c})}_{\text{discount factor}}\,\mathbf{p}_\mathrm{c} + \underbrace{\mathbb{1}_{[0,2n]}(\|\mathbf{p}_\sigma\|)}_{\text{stall}}\sqrt{1 - (1 - c_\mathrm{c})^2}\,\underbrace{\sqrt{\mu_w}\,\mathbf{y}_w}_{\text{under neutral selection } \mathcal{N}(\mathbf{0},\mathbf{C})} \qquad \text{path for } \mathbf{C}$$

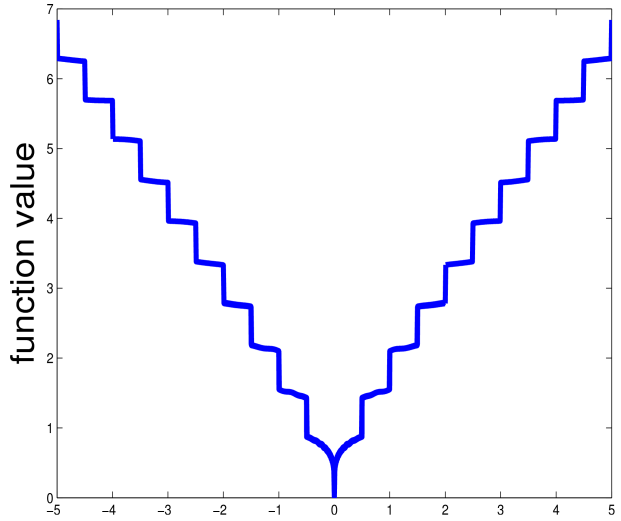
$$\mathbf{C} \leftarrow \underbrace{(1 - c_1 - c_\mu)}_{\text{discount factor}}\,\mathbf{C} + c_1\,\underbrace{\mathbf{p}_\mathrm{c}\,\mathbf{p}_\mathrm{c}^\mathrm{T}}_{\text{rank one}} + c_\mu\,\underbrace{\sum_{i=1}^{\lambda} \widehat{w}_{\rho(i)}\,\mathbf{y}_i\mathbf{y}_i^\mathrm{T}}_{\text{rank } \mu} \qquad \text{update } \mathbf{C}$$

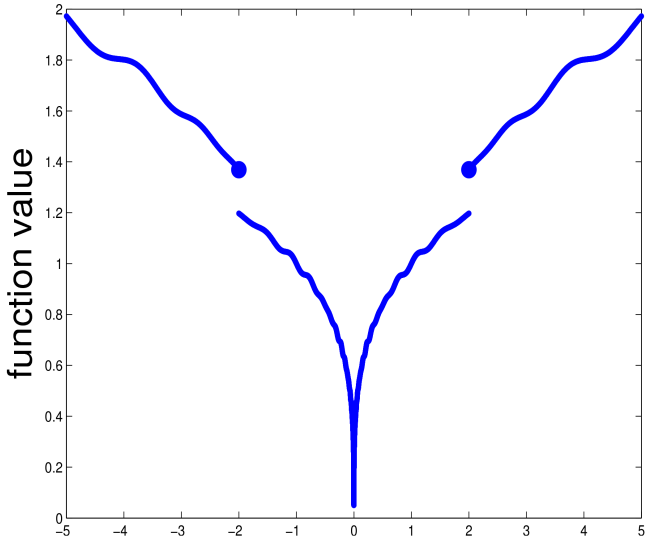
Design principles applied for CMA-ES

- Minimal prior assumptions
  - stochastic helps, maximum entropy distribution
  - Improvement only by selection of solutions
- 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* $\{f : \mathbb{R}^n \to \mathbb{R} \mid f = g \circ h \text{ with } g \text{ strictly increasing}\}$ *is an equivalence class with indistinguishable search trace.*

➡ derivative *and function-value* free

Role of invariance

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

- Invariance increases the value of theoretical claims or empirical observations, for example on the performance of an optimization algorithm
- Observed or proven results equivalently hold on a class of functions

invariance introduces equivalence classes

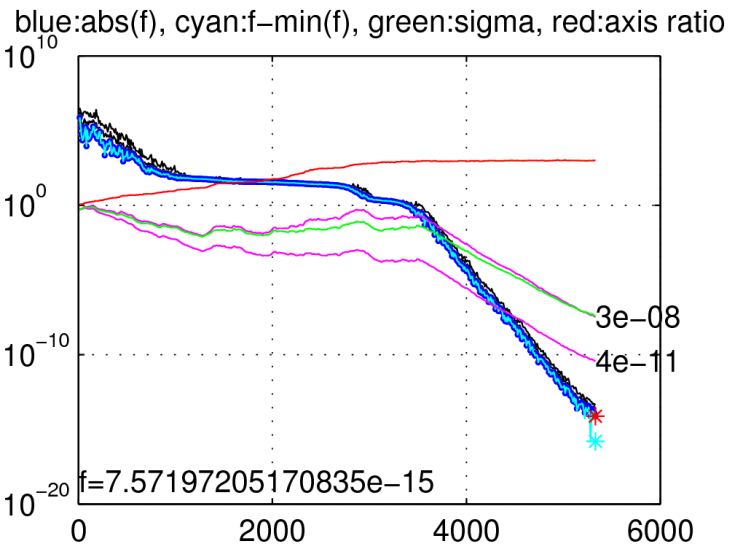
...experimental validation...

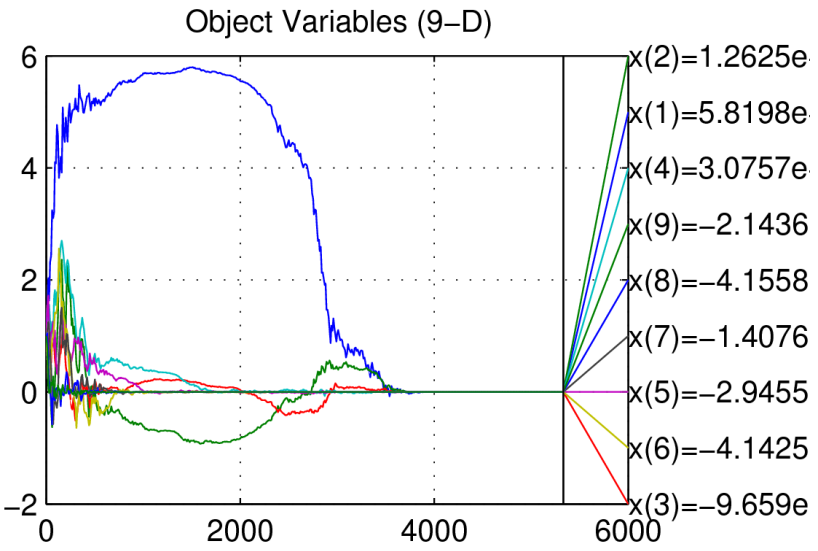
A simple unimodal test function

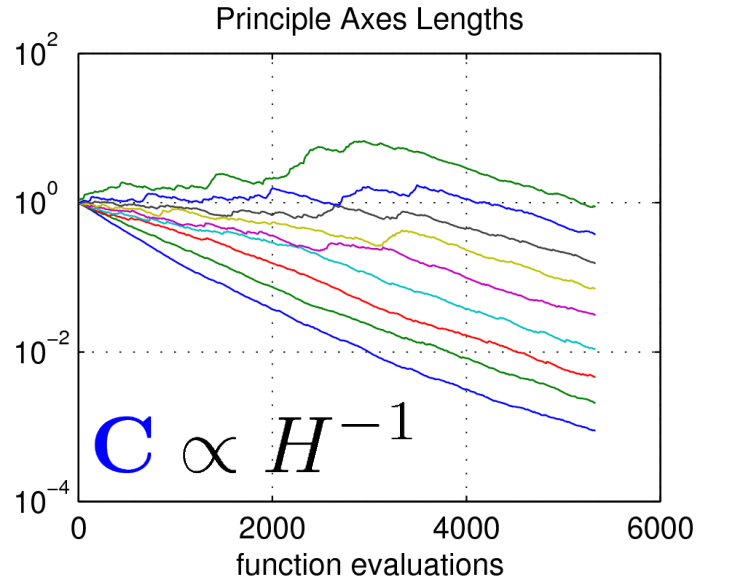
$$f(x) = g\left(\frac{1}{2} x^T H x\right)$$

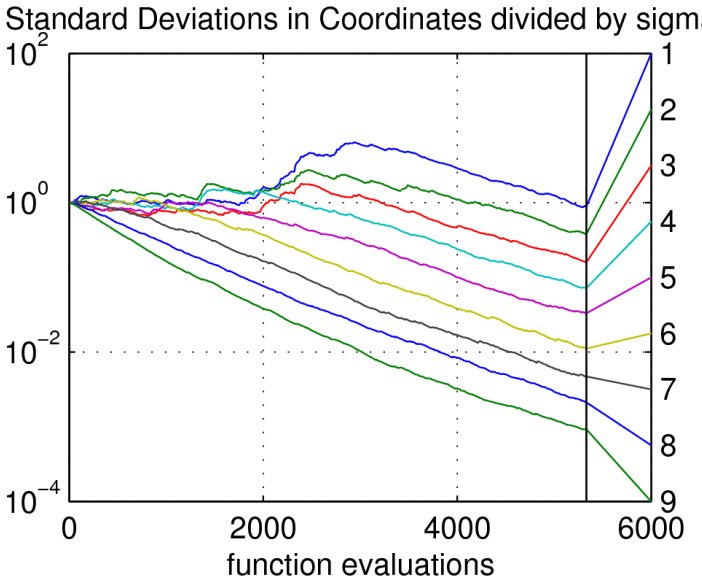
- for different strictly monotonic (i.e. order-preserving) $g : \mathbb{R} \to \mathbb{R}$
- with uniform eigenspectrum of the Hessian H
- with different condition numbers of H (ratio between largest and smallest eigenvalue) between one and 10^{10}
- in dimension 9 and 20

Experimentum crucis







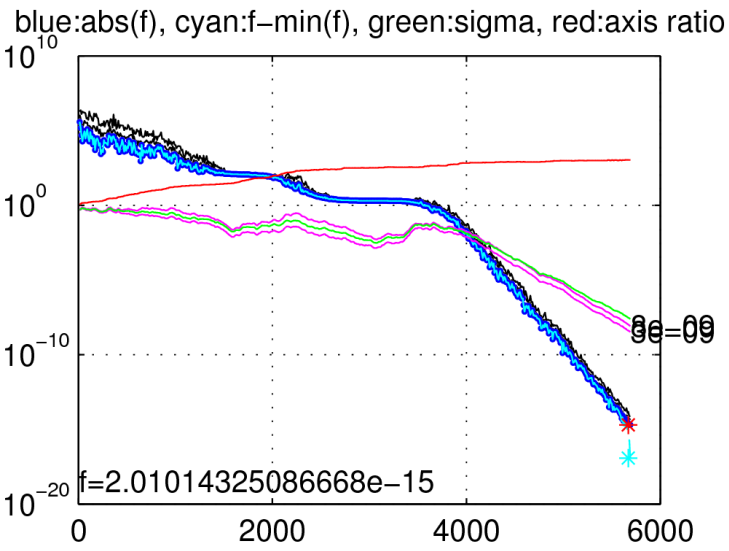


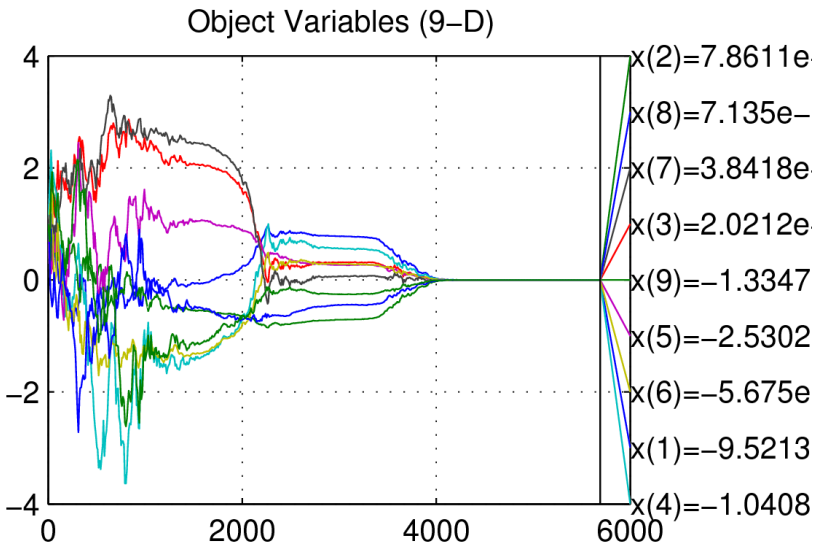
$$f(x) = \sum_{i=1}^{n} \alpha_i x_i^2$$

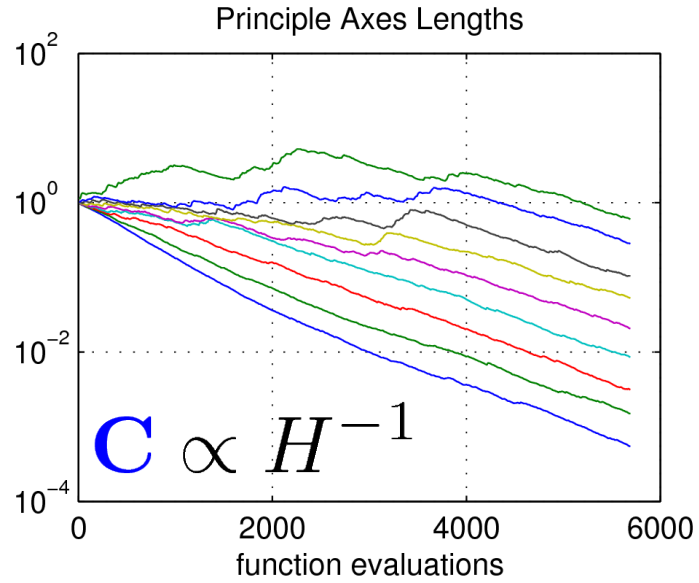
$$\alpha_i = 10^{6\frac{i-1}{n-1}}$$

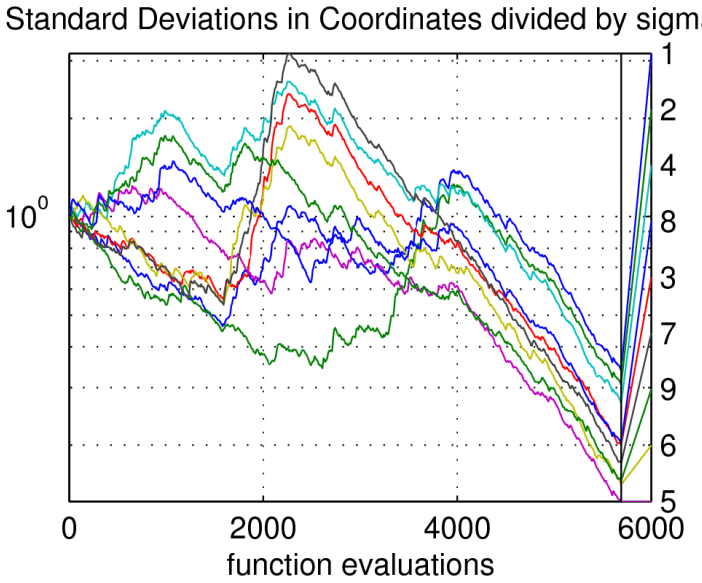
without covariance matrix adaptation it takes 1000 times longer to reach $f = 10^{-10}$

Experimentum crucis









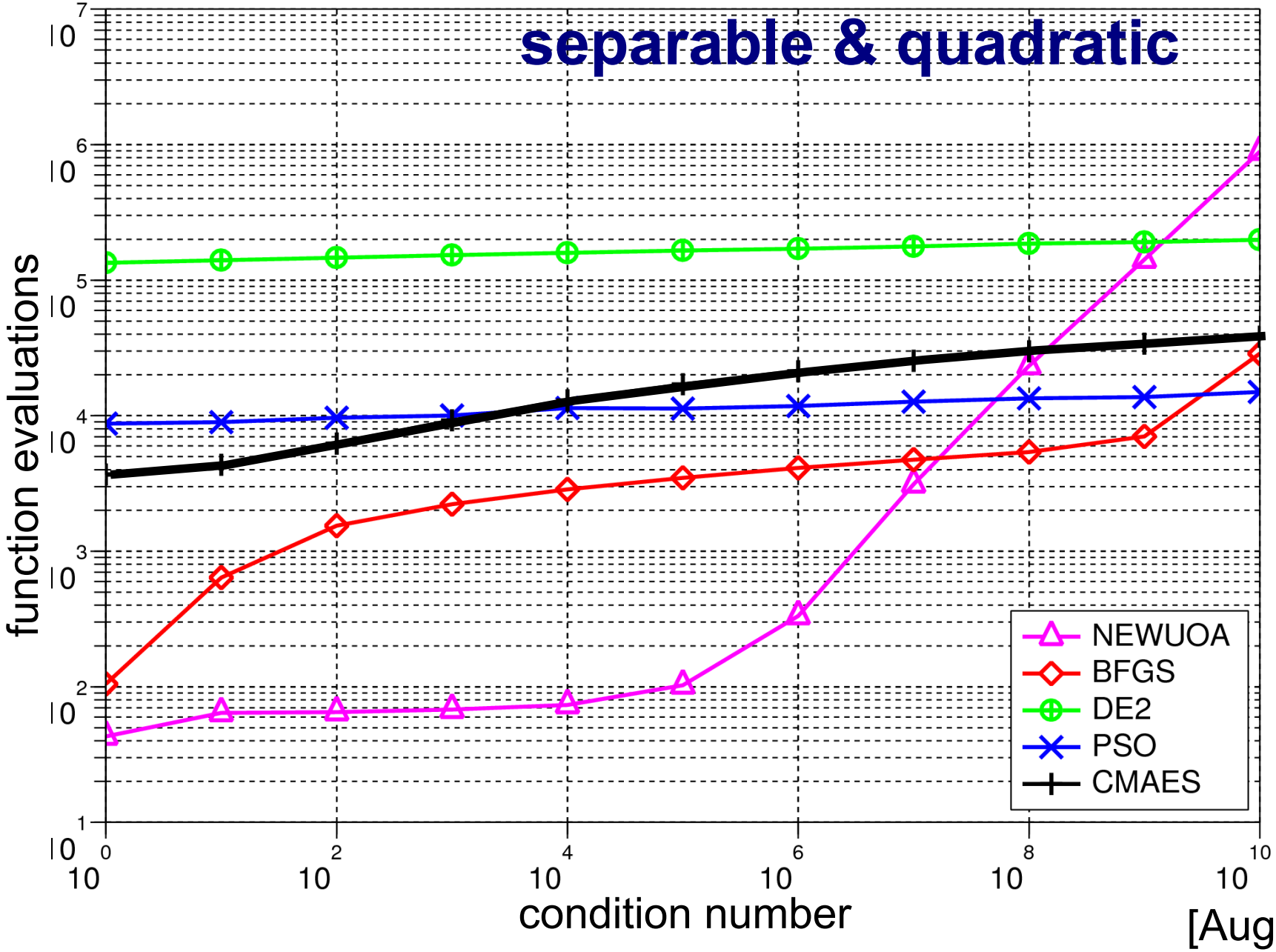
$$f(x) = \sum_{i=1}^{n} \alpha_i y_i^2$$

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

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

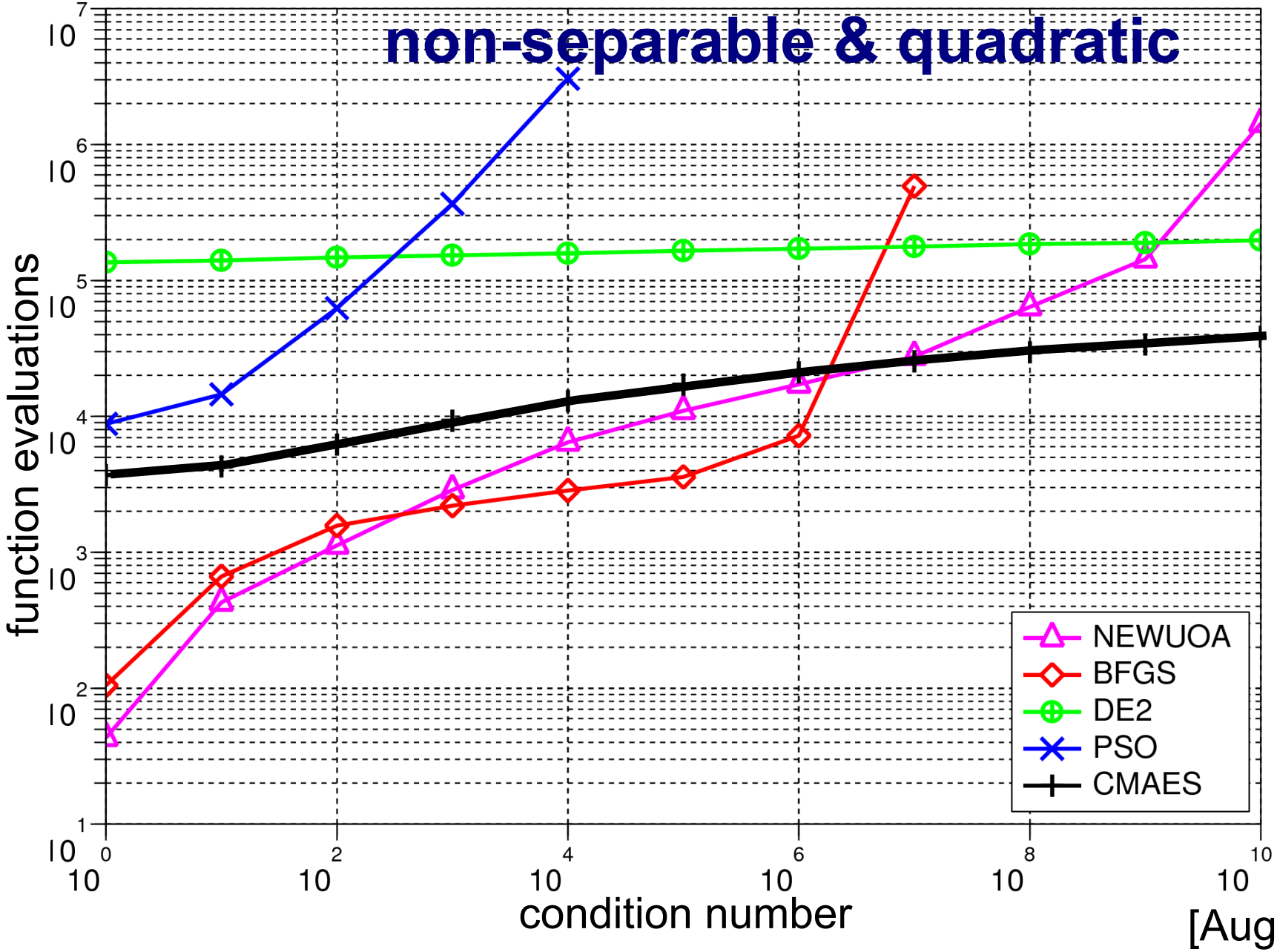
without covariance matrix adaptation it takes 1000 times longer to reach $f = 10^{-10}$

Runtime versus condition number



[Auger et al 2009]

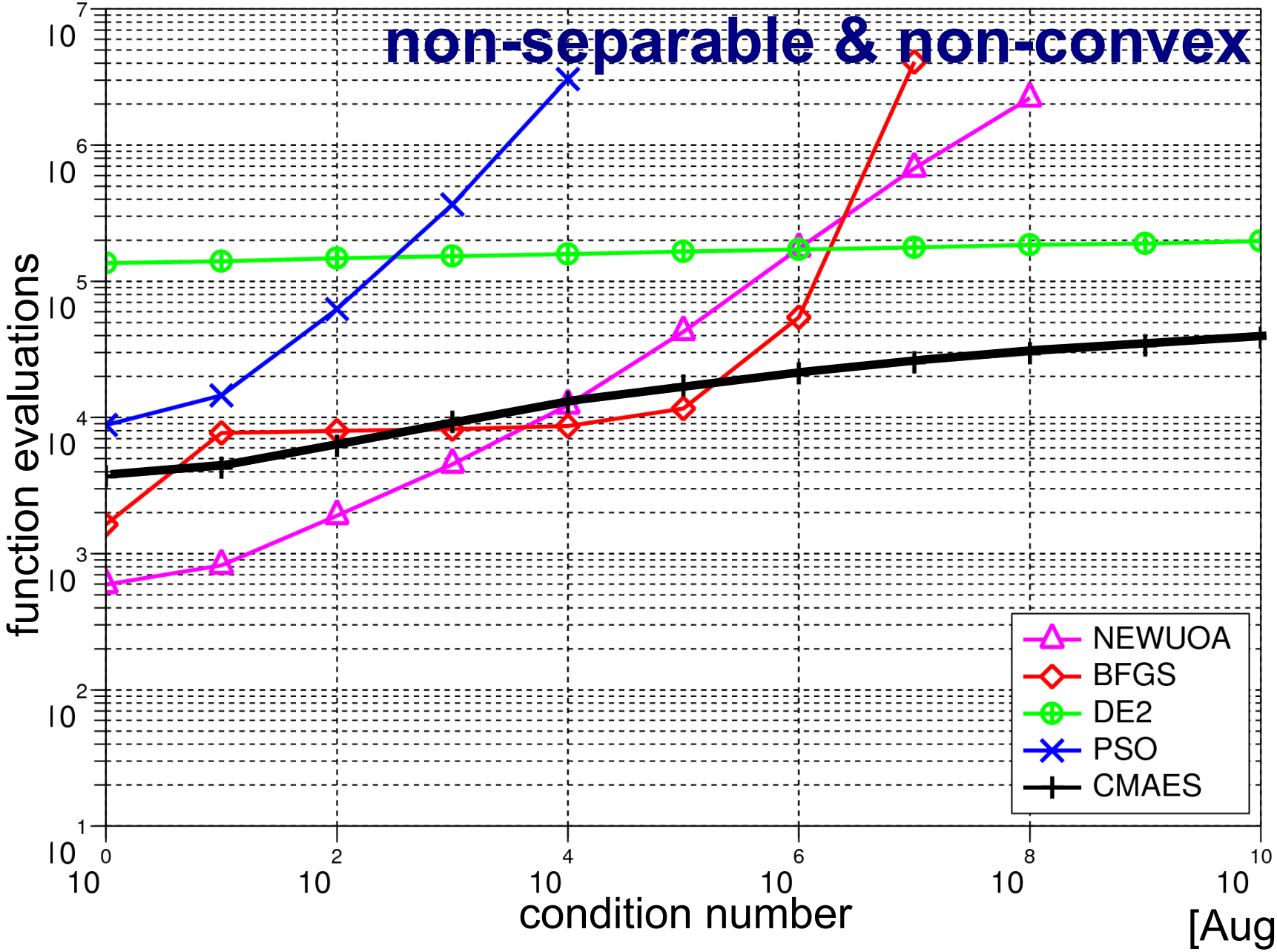
Runtime versus condition number



2

[Auger et al 2009]

Runtime versus condition number



[Auger et al 2009]

COCO/BBOB test environment

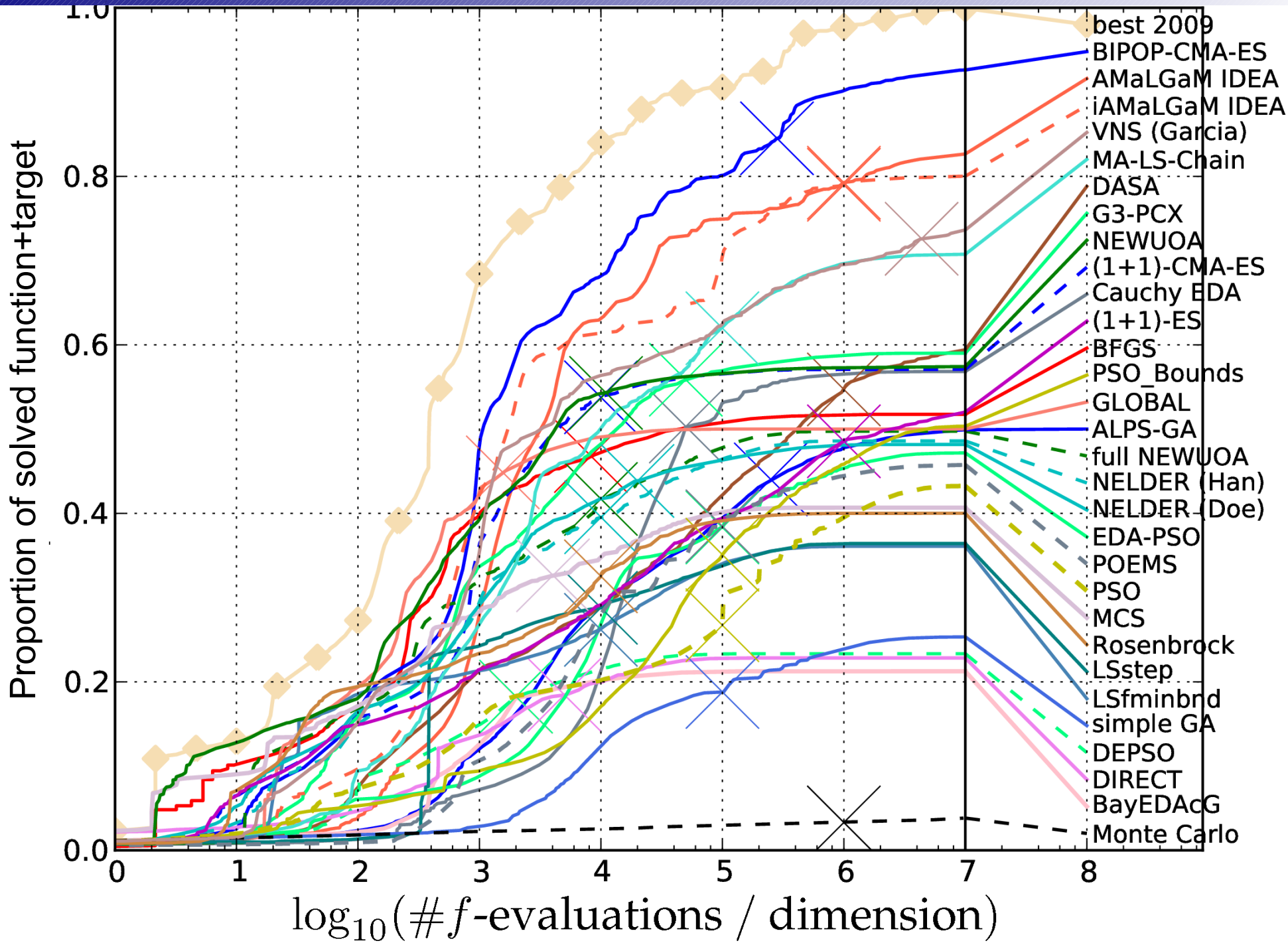
COCO: COmparing Continuous Optimizers

BBOB: Black-Box Optimization Benchmarking

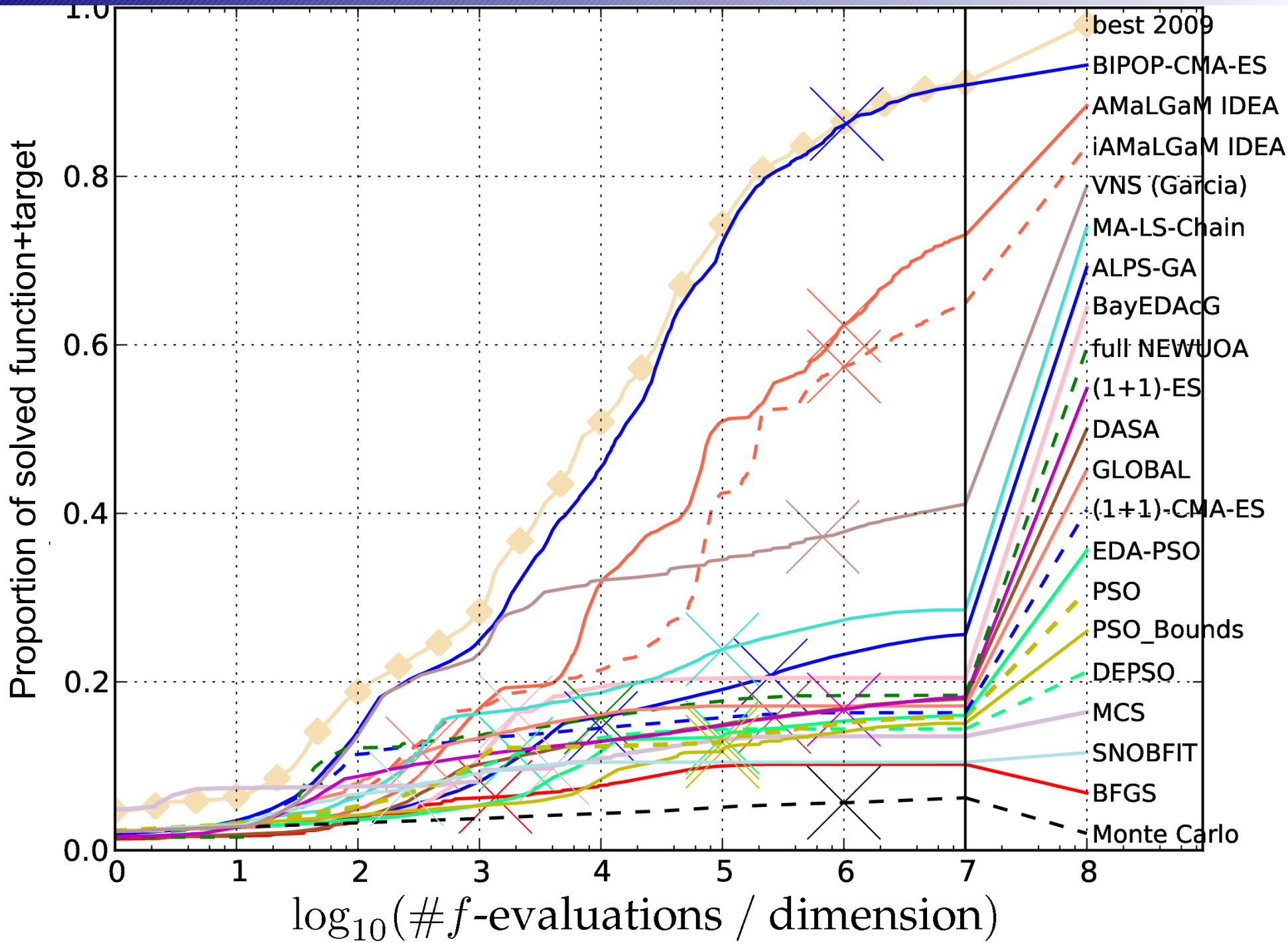
- 24 test functions with a wide range of difficulties
- 30 noisy functions

30+ algorithms have been tested:

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



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



Questions?

Furious activity is no substitute for understanding.

-- Albert Einstein

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

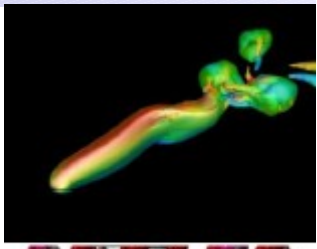
Limitations of CMA-ES

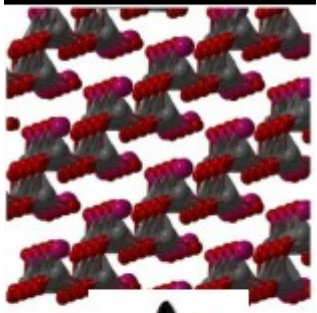
- internal CPU-time: $10^{-8}n^2$ seconds per function evaluation on a 2GHz PC, tweaks are available
  1 000 000 f-evaluations in 100-D take 100 seconds *internal* CPU-time

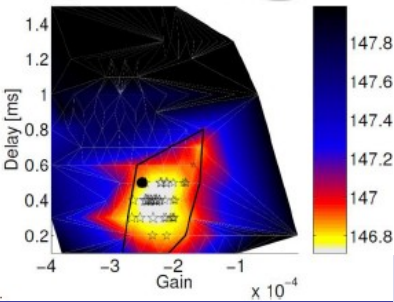
- better methods are presumably available in case of
  - partly separable problems
  - specific problems, for example with cheap gradients
    specific methods
  - small dimension ($n \ll 10$)
    for example Nelder-Mead
  - small running times (number of f-evaluations $\ll 100n$)
    model-based methods

More selected applications

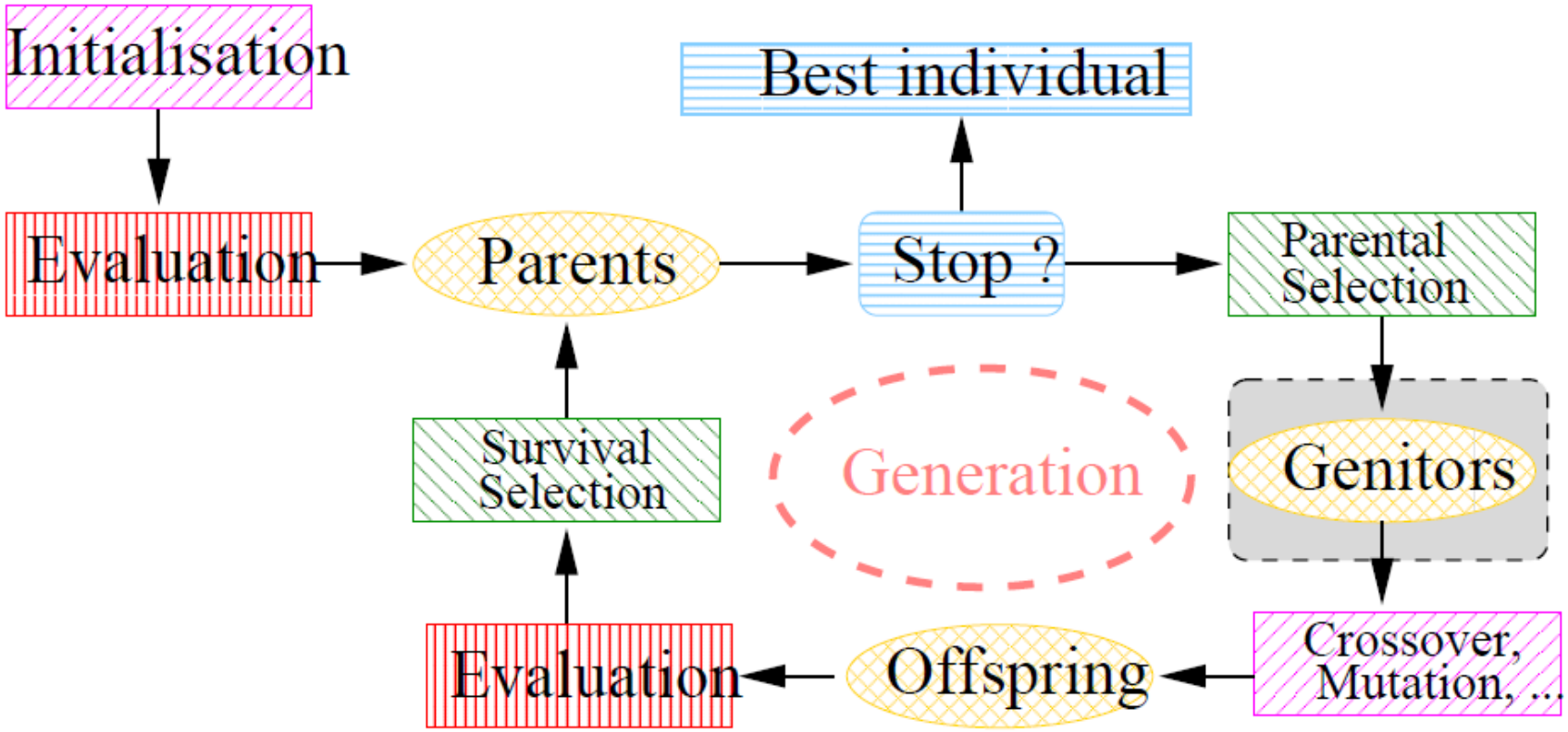
- Swimming fish simulation [Kern et al 2007]
  computational flow simulation,
  motion control
- Crystal structure prediction [Glass et al 2006]
  specialized algorithm: encoding, operators etc.
  new structure of $CaCO_3$ above 137GPa predicted
  and subsequently confirmed in experiment
- Modelling of volcanic magma [Halter et al 2006]
  bilevel energy optimization
- Space launcher design to maximize the payload per EUR [Collange et al 2010]
  for Ariane in collaboration with EADS Astrium
- Combustion control [Hansen et al 2009]
  real-time laboratory experiment
  in collaboration with Alstom







Evolutionary Algorithm Scheme



Stochastic operators Representation dependent

Darwinian Evolution Engine (can be stochastic or deterministic)

Main CPU cost

Checkpointing: stopping criterion, statistics, updates, ...

CMA-ES is 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 LRI source code download page