Numerical Optimization with CMA-ES: From Theory to Practice and from Practice to Theory

Anne Auger

Optimization and Machine Learning Team (TAO)
INRIA Saclay-Ile-de-France

Mascot-Num
8 - 10 April 2015

Note: some slides are taken from the CMA-ES tutorial given by A. Auger and N. Hansen at GECCO 2014

Zero Order Numerical / Continuous Black-Box Optimization

Optimize $f : \mathbb{R}^n \mapsto \mathbb{R}$

Zero[th] order method + Black-Box setting

Gradients not available or not useful

Knowledge about the problem encoded within the black box but not exploited by the algorithms

Cost = # calls to the black-box (f-calls)

Example of Numerical Black-box Problem
Optimization of Oil Well Placement

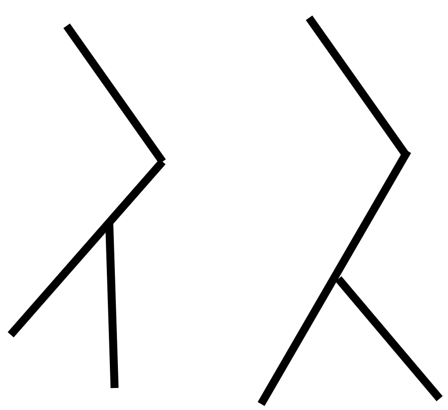
Z. Bouzarkouna PhD thesis, collab IFP

$x \in \mathbb{R}^n$

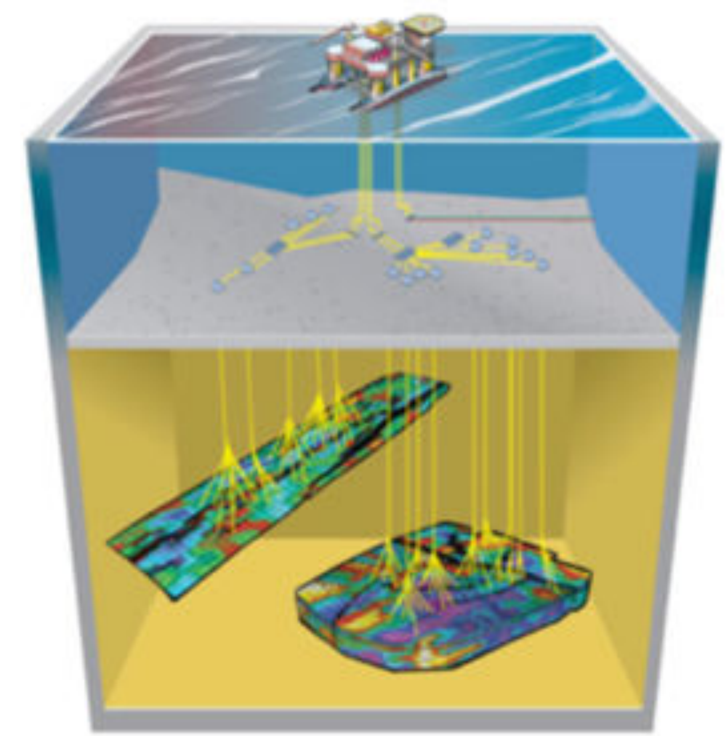
commercial solver
(source code not available)
predict quantity of oil for
given set of wells

NPV (in \$)

well configurations



expensive optimization
gradient not provided



Landscape of Numerical Zero Order Methods

implicit assumption: $\#$ f-evals $\geq 100n$

Derivative-free optimization (deterministic)

mathematical programming

Trust-region methods (NEWUOA, BOBYQA) [Powell 2006, 2009]

Simplex downhill [Nelder & Mead 1965]

Pattern search [Hook and Jeeves 1961]

Quasi-Newton with estimation of gradient (BFGS) [Broyden et al. 1970]

Stochastic (randomized) search methods

Evolutionary Algorithms (continuous domain)

- Differential Evolution [Storm & Price 1997]
- Particle Swarm Optimization [Kennedy & Eberhart 1995]
- **Evolution Strategies** [Rechenberg 1965, Hansen & Ostermeier 2001]
- Genetic Algorithms [Holland 1975, Goldberg 1989]

Simulated annealing [Kirkpatrick et al. 1983]

Simultaneous perturbation stochastic approximation (SPSA) [Spall 2000]

Landscape of Numerical Zero Order Methods

Derivative-free optimization (deterministic)

mathematical programming

Trust-region methods (NEWUOA, BOBYQA) [Powell 2006, 2009]

Simplex downhill [Nelc

“local” optimization

Pattern search [Hook and Jeeves 1961]

Quasi-Newton with estimation of gradient (BFGS) [Broyden et al. 1970]

Stochastic (randomized) search methods

Evolutionary Algorithms (continuous domain)

- Differential Evolution [Storm & Price 1997]
- Particle Swarm Opti

“global” optimization

- **Evolution Strategies** [Rechenberg 1965, Hansen & Ostermeier 2001]
- Genetic Algorithms [Holland 1975, Goldberg 1989]

Simulated annealing [Kirkpatrick et al. 1983]

Simultaneous perturbation stochastic approximation (SPSA) [Spall 2000]

CMA-ES in a nutshell
Covariance Matrix Adaptation Evolution Strategy

N. Hansen main driving force behind

Stochastic comparison-based algorithm

- Variable-metric method
- Robust local search
- Parameter-free algorithm

state-of-the-art stochastic continuous optimizer

Mature code available in C, Python, C++, Java, R, Matlab/Octave, Fortran

- available at https://www.lri.fr/~hansen/cmaes_inmatlab.html

algorithm: google Hansen + CMA-ES

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Which Typical Difficulties Need to be Addressed?

The algorithm does not know in advance the features/ difficulties of the optimization problem that need to be solved

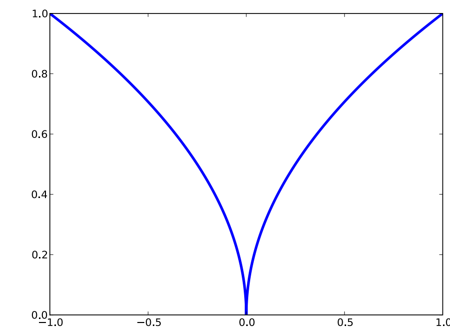
difficulties related to real-wold problems

Has to be ready to solve any of them

What Makes a Function Difficult to S

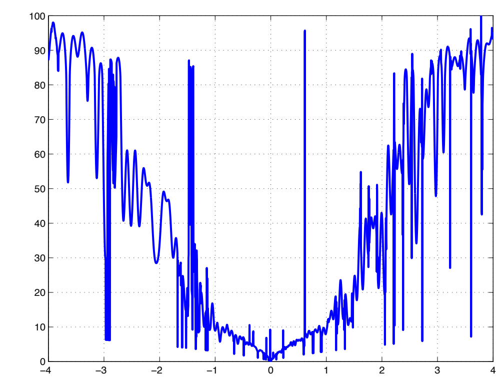
Why Comparison-based Stochastic S

non-linear, non-quadratic, non convex



ruggedness

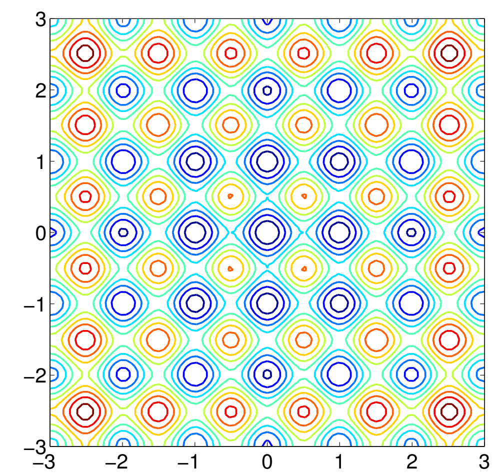
non-smooth, discontinuous, multi-modal, and/or noisy functions



dimensionality (size of the search space)

(considerably) larger than three

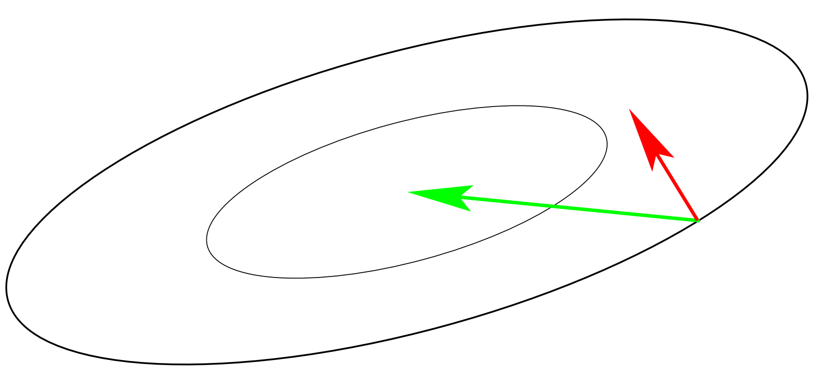
curse of dimensionality



non-separability

dependencies between the objective variables

ill-conditioning



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.

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 appear now as isolated points in a vast empty space.

Consequence: a search policy (e.g. exhaustive search) that is valuable in small dimensions might be useless in moderate or large dimensional search spaces.

Non-separability

Definition (Separable Problem)

A function f is separable if

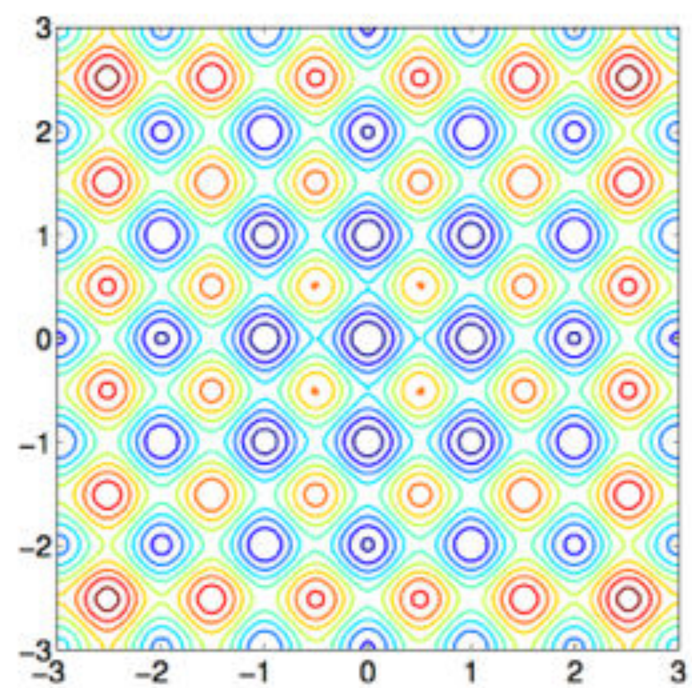
$$\arg\min_{(x_1,\dots,x_n)} f(x_1,\dots,x_n) = \left(\arg\min_{x_1} f(x_1,\dots),\dots,\arg\min_{x_n} f(\dots,x_n)\right)$$

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

Example: Additively decomposable functions

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

Rastrigin function
$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$



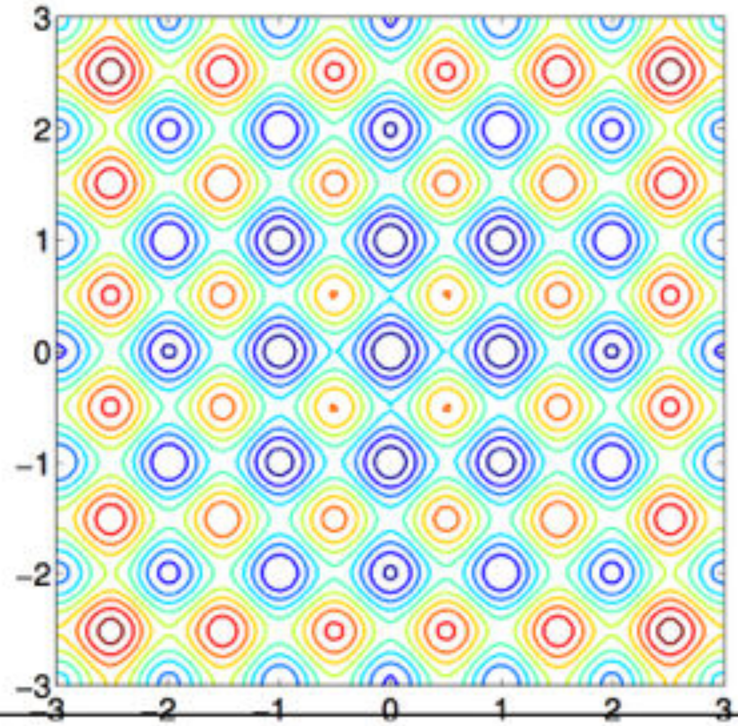
Non-separability (cont.)

Building a non-separable problem from a separable one [1,2]

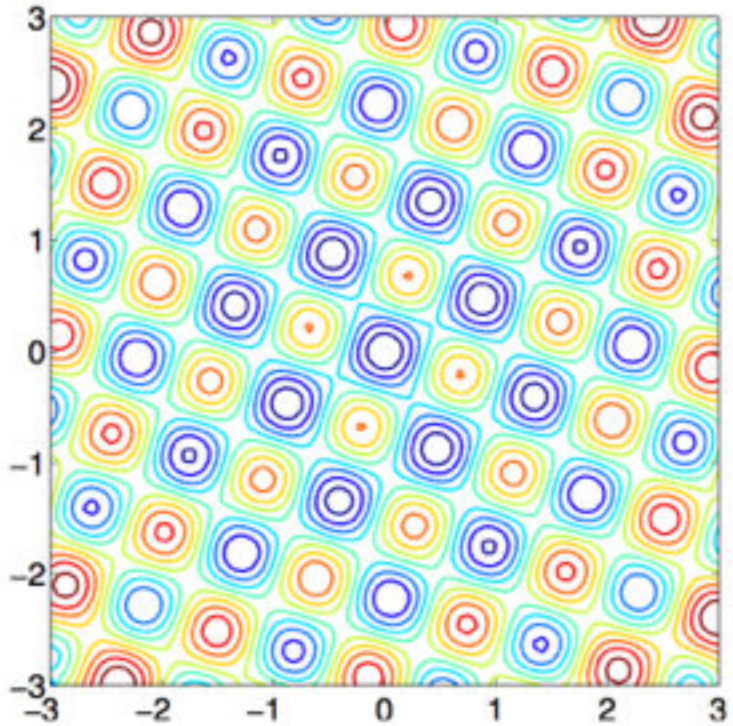
Rotating the coordinate system

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

R rotation matrix



$\xrightarrow{\mathbf{R}}$



[1] 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

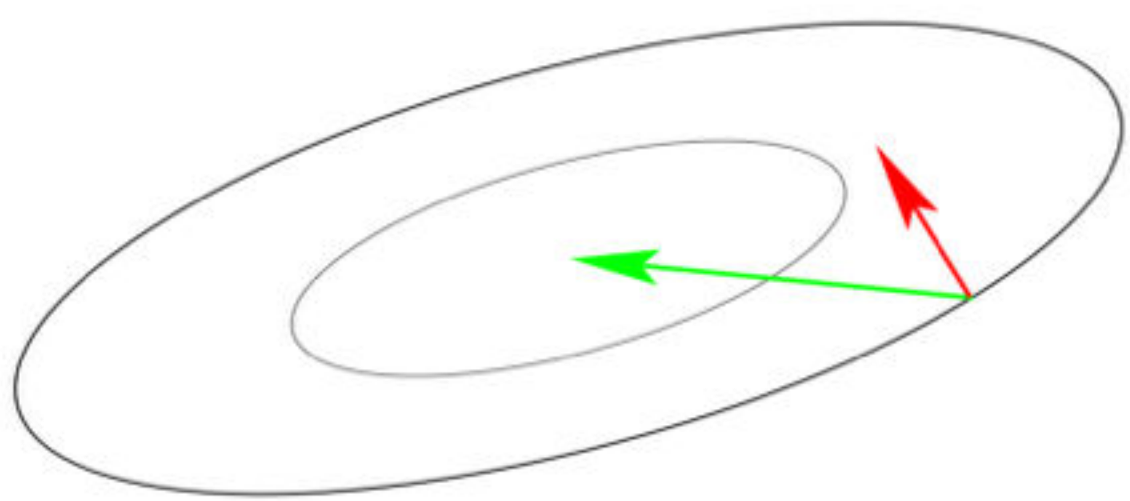
[2] Salomon (1996). "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

Ill-conditioned Problems
Curvature of level sets

Consider the convex-quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^T\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}^*) = \frac{1}{2}\sum_i h_{i,i}\,(x_i-x_i^*)^2 + \frac{1}{2}\sum_{i\neq j} h_{i,j}\,(x_i-x_i^*)(x_j-x_j^*)$$

$\boldsymbol{H}$ is Hessian matrix of f and symmetric positive definite



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

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

Ill-conditioning means squeezed level sets (high curvature).
Condition number equals nine here. Condition numbers up to 10^{10} are not unusual in real world problems.

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

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

CMA-ES High-level Template

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

Initialize distribution parameters θ, set population size λ
While not terminate

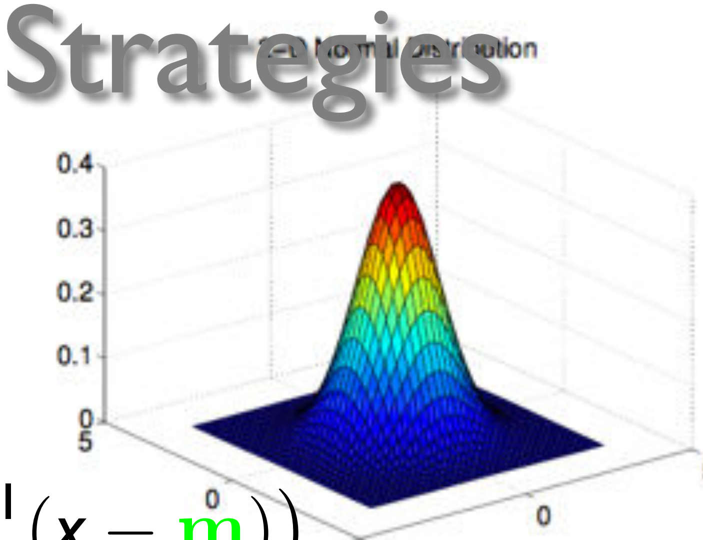
1. Sample distribution $p_\theta(x) : x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on f
3. Update parameters $\theta \leftarrow F(\theta, x_1, \ldots, x_\lambda, f(x_1), \ldots, f(x_\lambda))$

Everything depends on p_θ and F

p_θ proba. distribution encodes the belief where good solutions are located

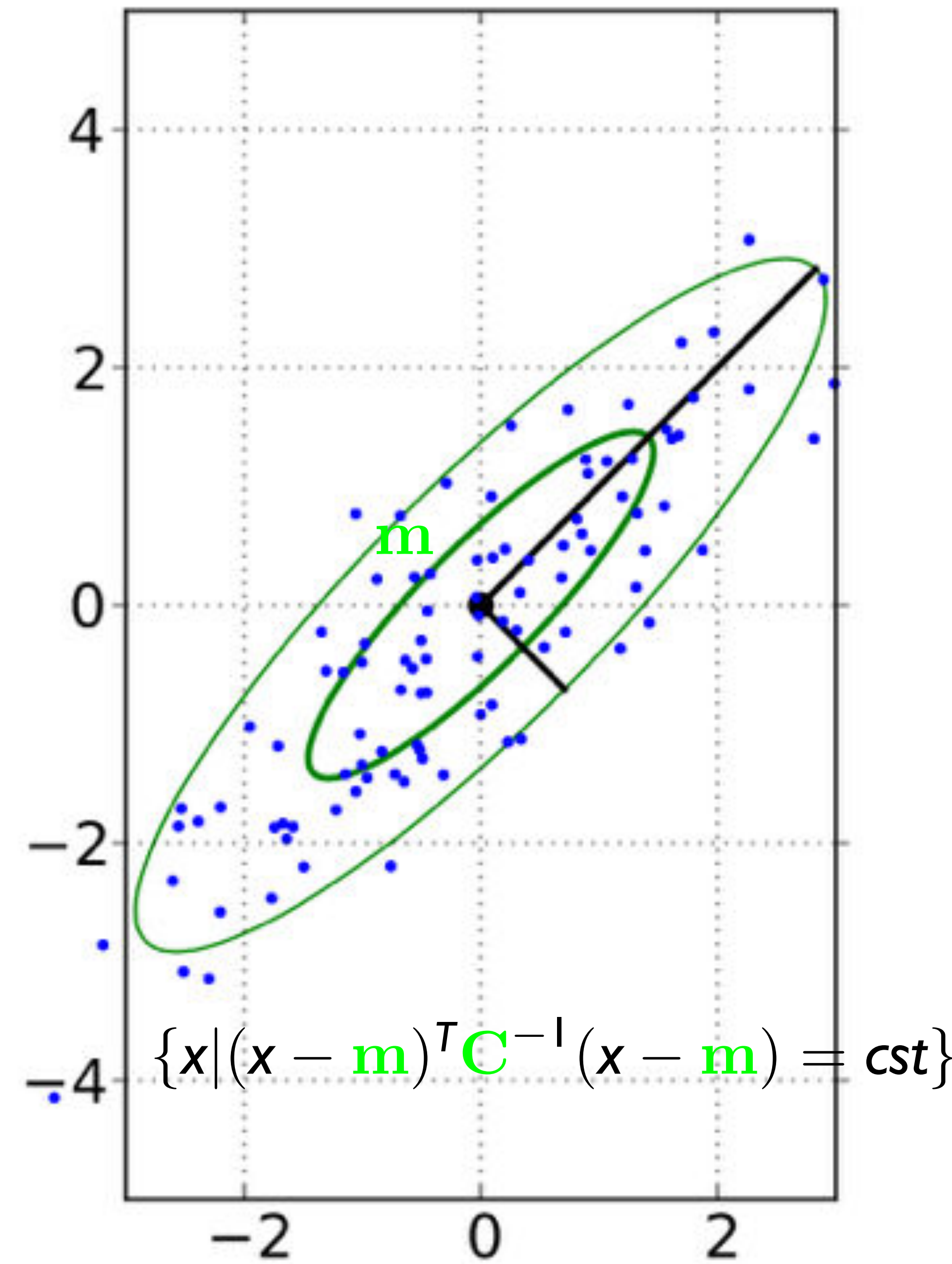
F should drive the algorithm to converge towards some optima of the objective function

Sampling Distribution in Evolution Strategies
Multivariate Normal Distributions



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

density : $p_{\theta := (\mathbf{m}, \mathbf{C})}(x) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{C}|}} \exp\left(-\frac{1}{2}(x - \mathbf{m})^T \mathbf{C}^{-1} (x - \mathbf{m})\right)$



$\mathbf{m} \in \mathbb{R}^n$ favorite (incumbent) solution at a given iteration

σ overall scaling - step-size of the algorithm

$\mathbf{C}$ Symmetric definite positive matrix - encodes the geometric shape of the distribution

Variable metric method: the multivariate normal distribution encodes the underlying metric of CMA-ES

CMA-ES High-level Template

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

Initialize distribution parameters θ, set population size λ
While not terminate

1. Sample distribution $p_\theta(x) : x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on f
3. Update parameters $\theta \leftarrow F(\theta, x_1, \ldots, x_\lambda, f(x_1), \ldots, f(x_\lambda))$

Next:

Insights into how m, σ, C are updated in CMA-ES

Updating the mean vector m

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

Non-elitist selection and intermediate (weighted) recombination

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

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

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

where

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

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

Invariance under Monotonically Increasing Transformations of f

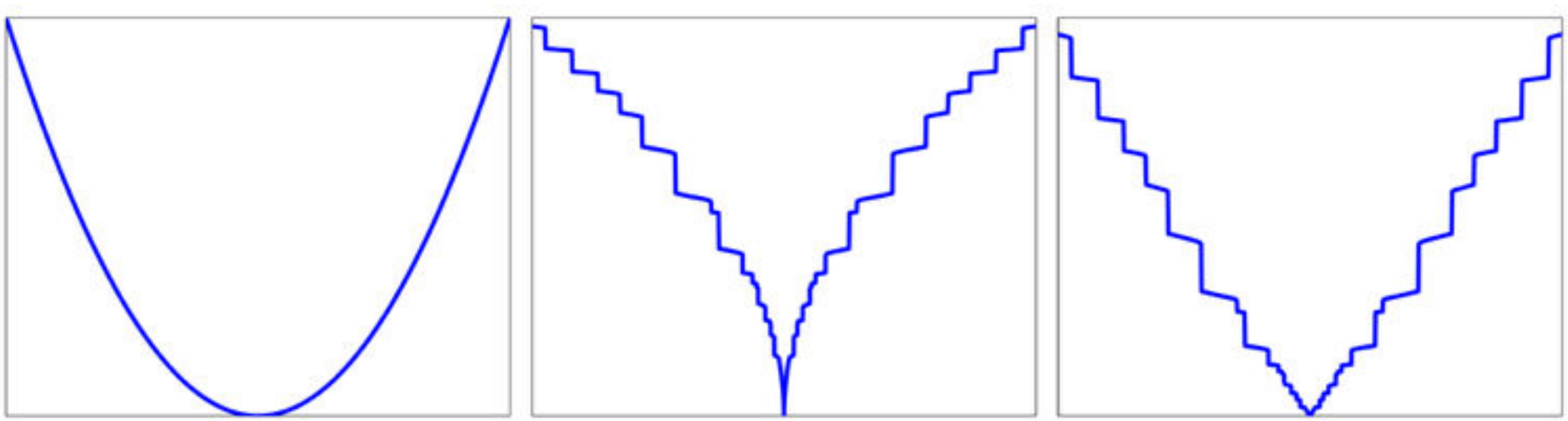
Comparison-based algorithm:

Update of all parameters uses only ranking of solutions

$$f(\mathbf{x}_{1:\lambda}) \leq f(\mathbf{x}_{2:\lambda}) \leq \ldots \leq f(\mathbf{x}_{\lambda:\lambda})$$

Same ranking on f or $g \circ f$ if $g : \mathbb{R} \mapsto \mathbb{R}$ strictly increasing

$$g \circ f(\mathbf{x}_{1:\lambda}) \leq g \circ f(\mathbf{x}_{2:\lambda}) \leq \ldots \leq g \circ f(\mathbf{x}_{\lambda:\lambda})$$



Invariance to strict. increasing transformations of f

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

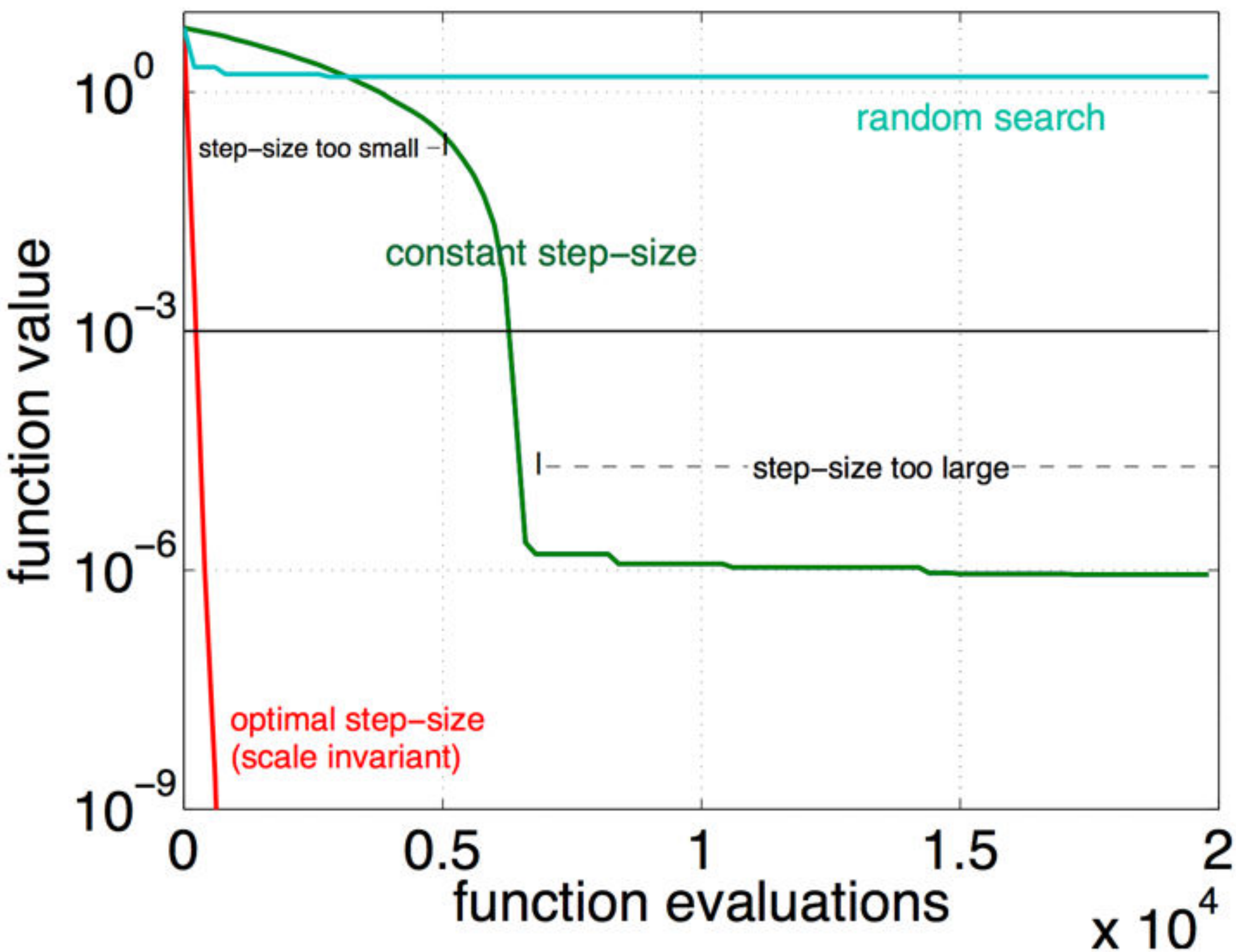
Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Updating the Step-size
Why step-size control?



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

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

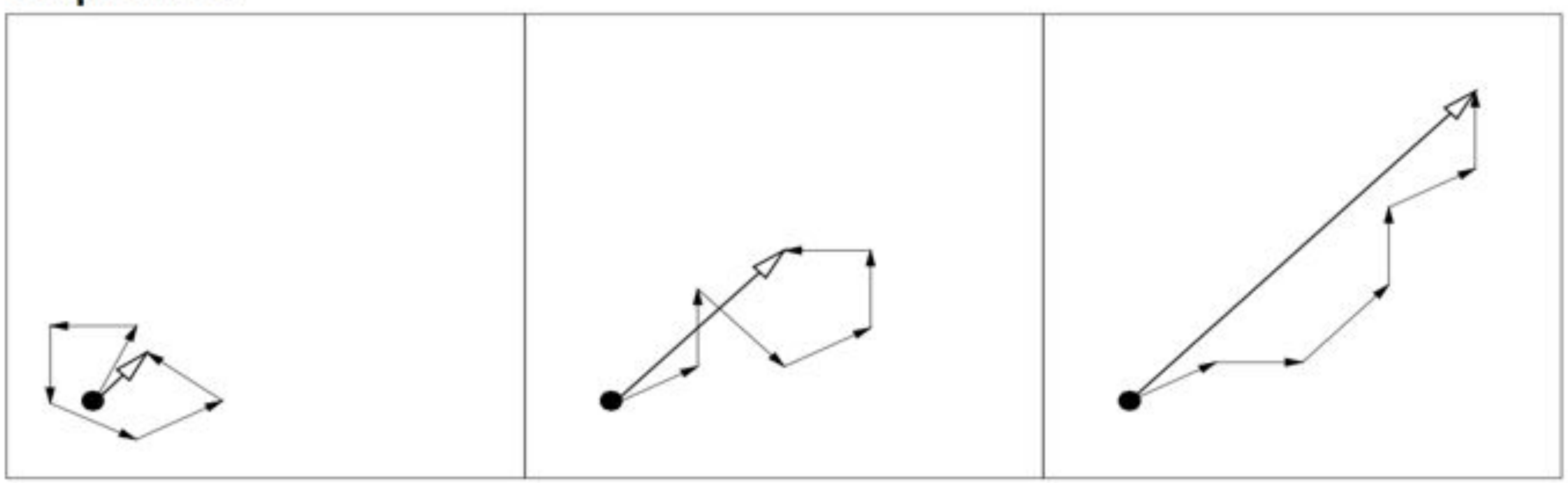
Path Length Control (CSA)

The Concept of Cumulative Step-Size Adaptation

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

Measure the length of the *evolution path*

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



$\Downarrow$
decrease σ

$\Downarrow$
increase σ

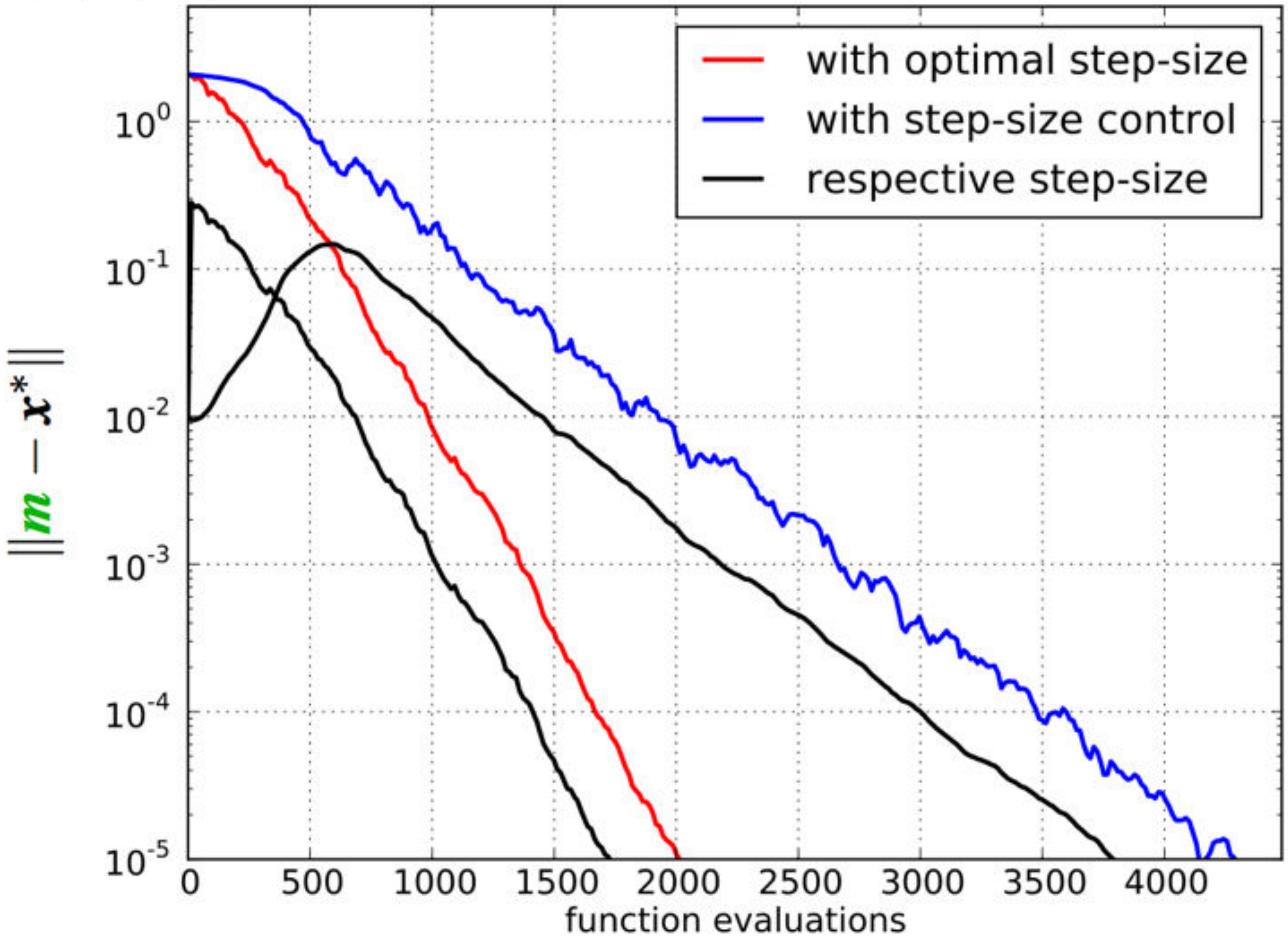
Path Length Control (CSA)

The Equations

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

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

$(5/5, 10)$-CSA-ES, default parameters



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

in $[-0.2, 0.8]^n$
for $n = 30$

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

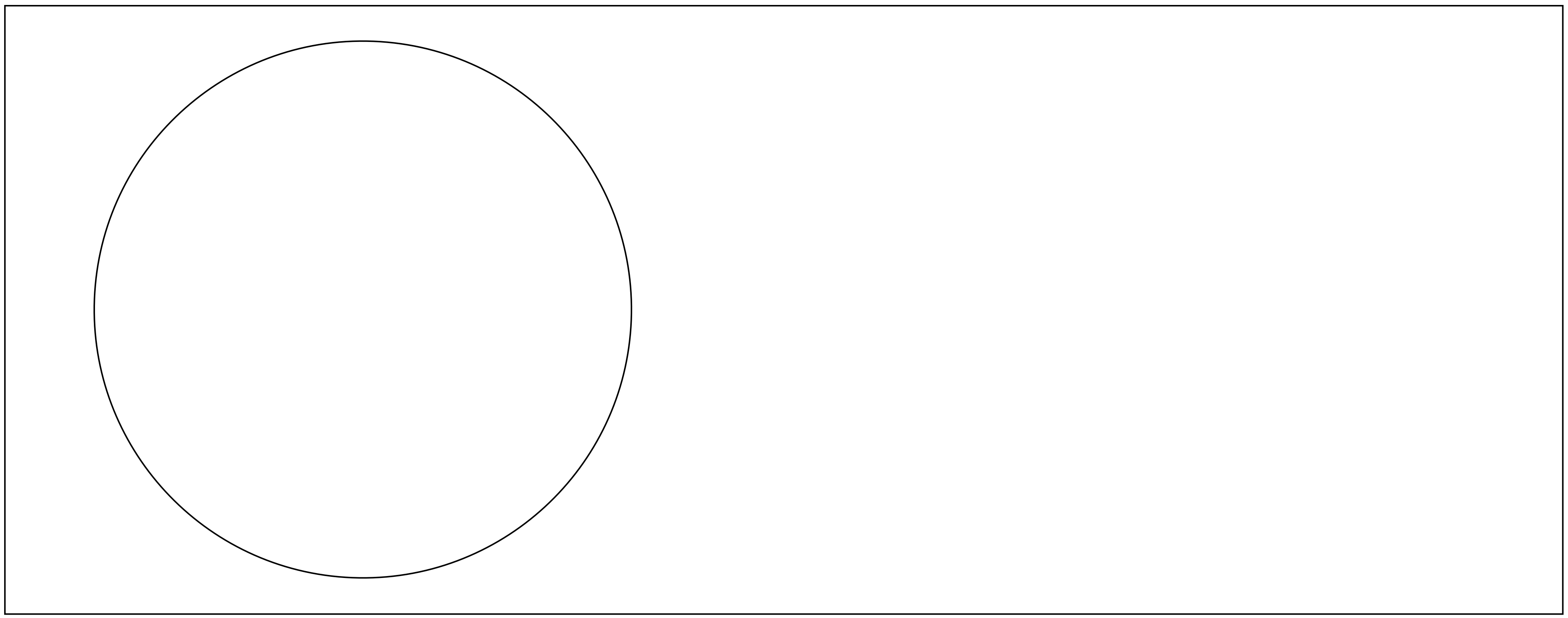
Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Covariance Matrix Adaptation

Rank-One Update

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



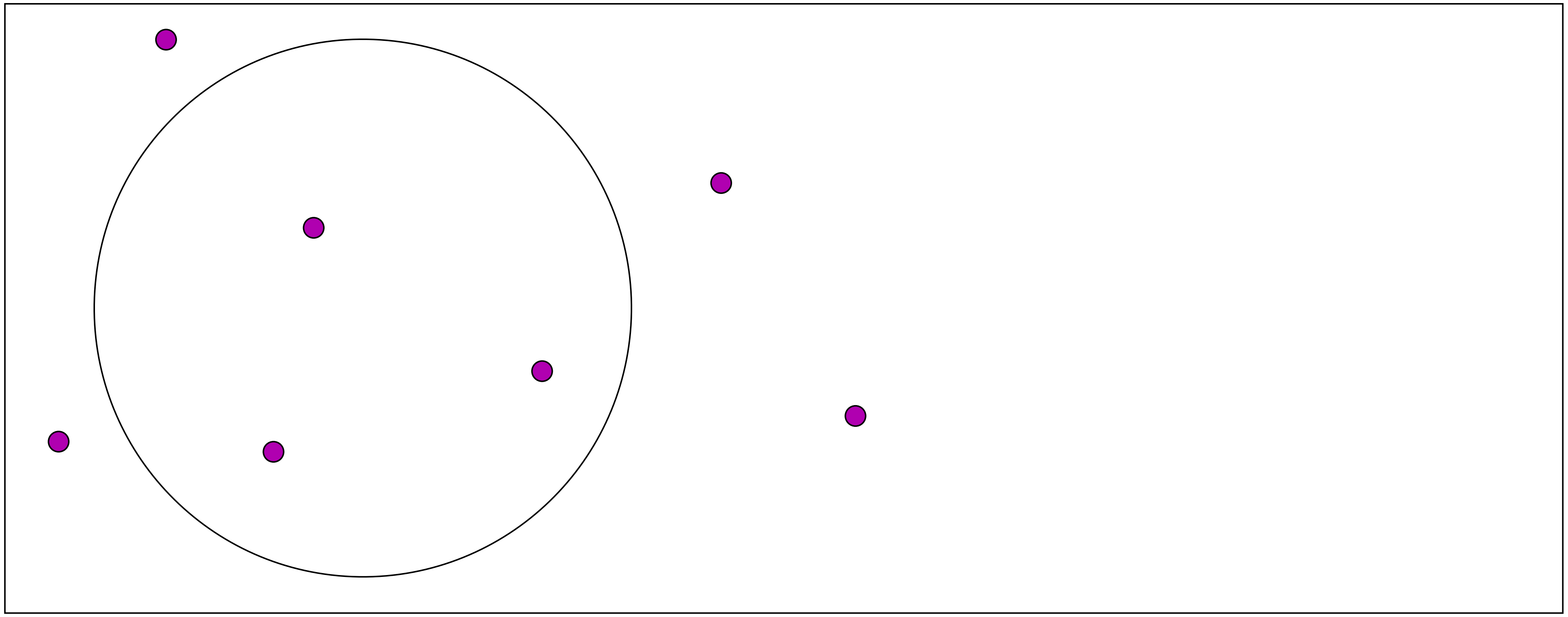
initial distribution, $\mathbf{C} = \mathbf{I}$

. . . equations

Covariance Matrix Adaptation

Rank-One Update

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



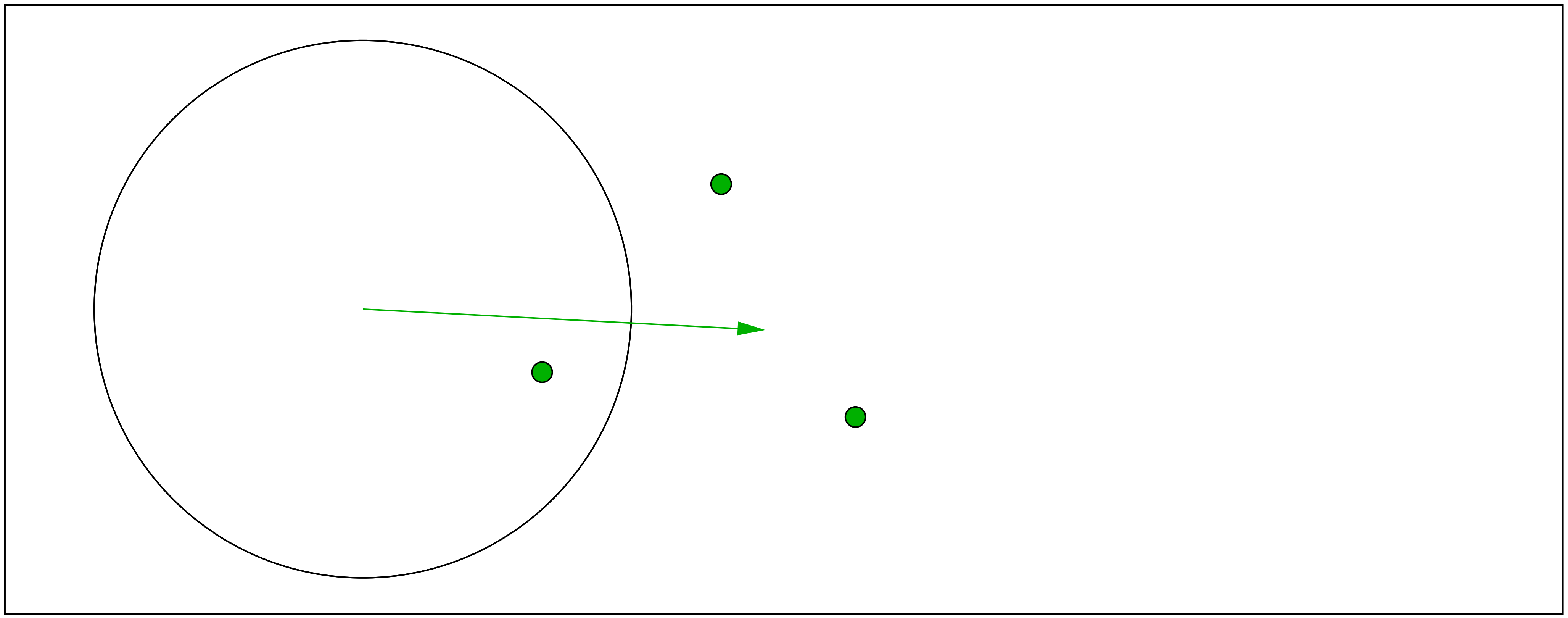
initial distribution, $\mathbf{C} = \mathbf{I}$

... equations

Covariance Matrix Adaptation

Rank-One Update

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



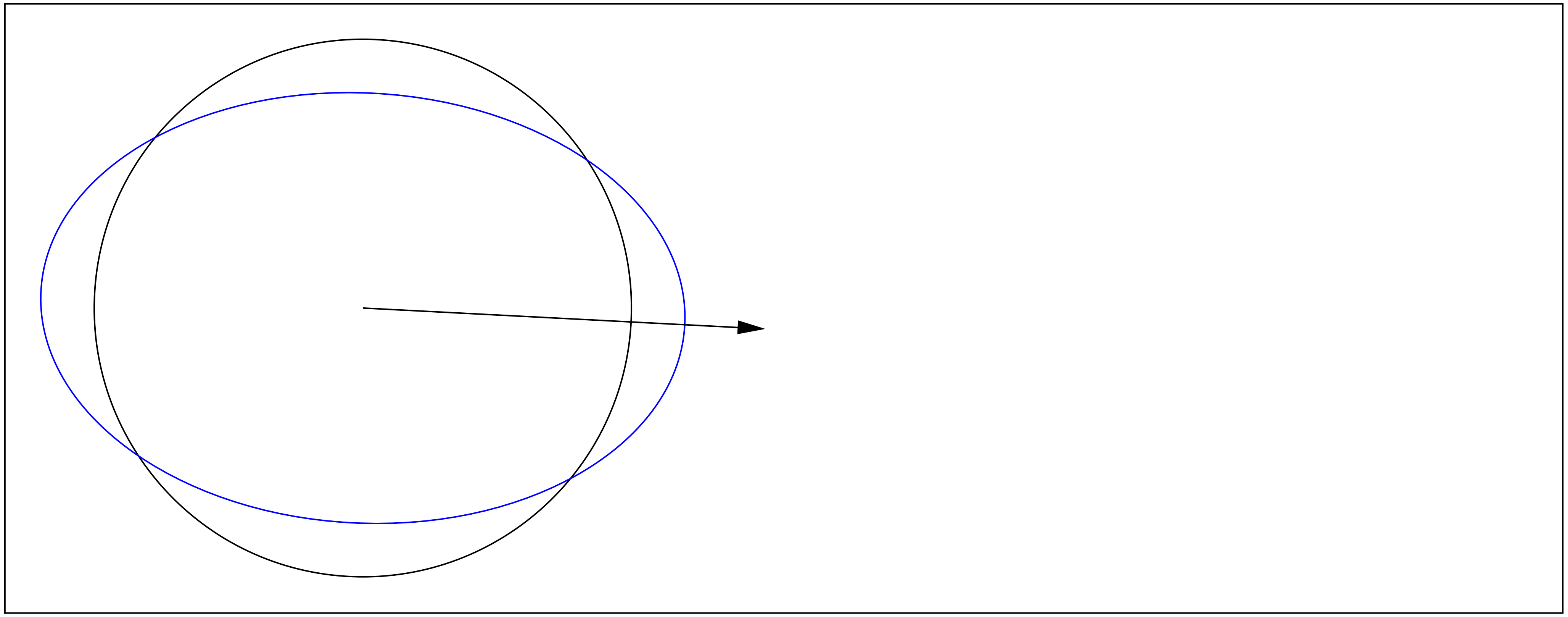
$\boldsymbol{y}_w$, movement of the population mean $\boldsymbol{m}$ (disregarding σ)

. . . equations

Covariance Matrix Adaptation

Rank-One Update

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



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

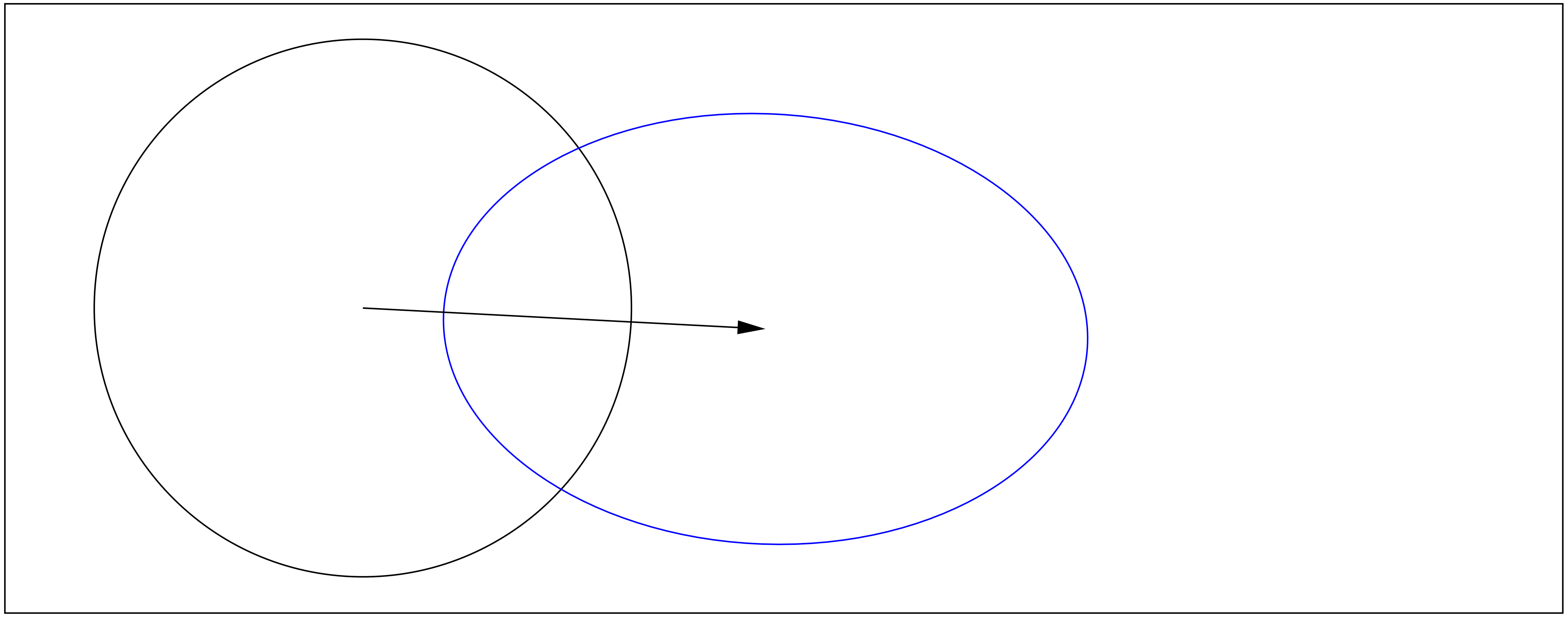
$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$

...equations

Covariance Matrix Adaptation

Rank-One Update

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

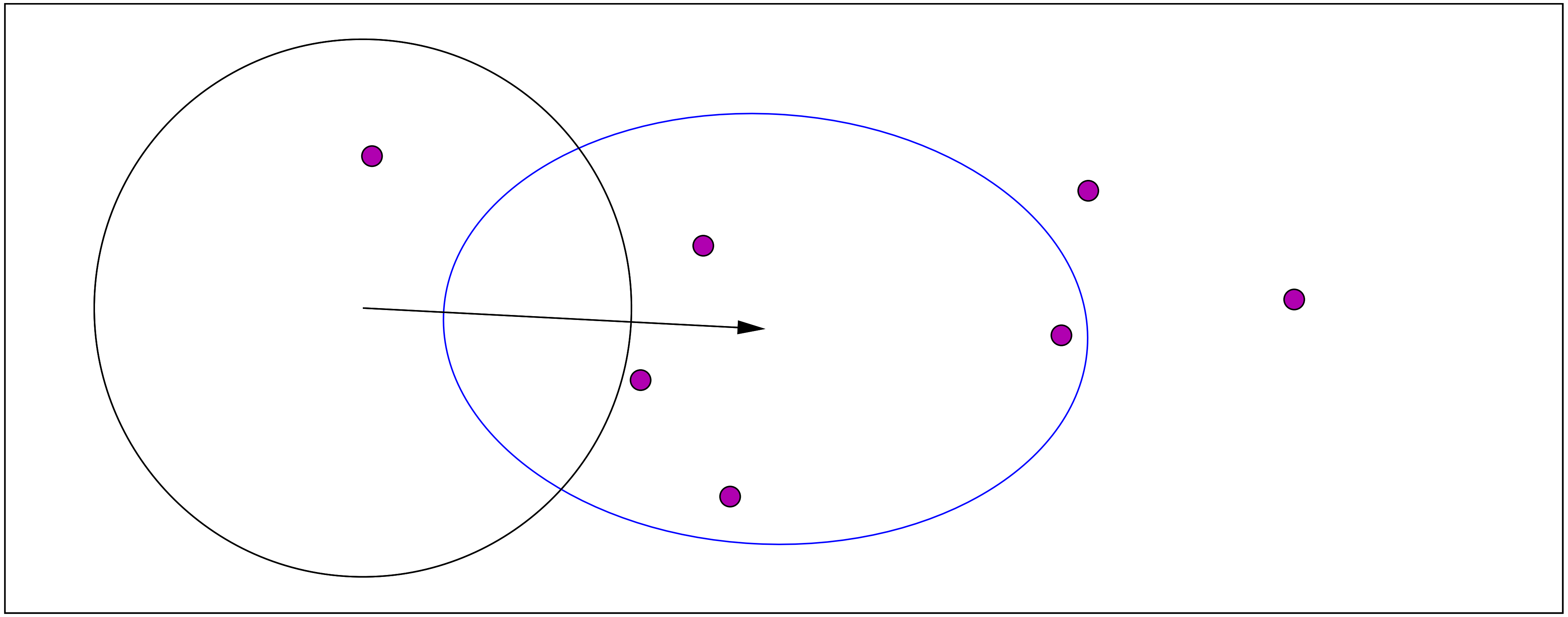


new distribution (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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

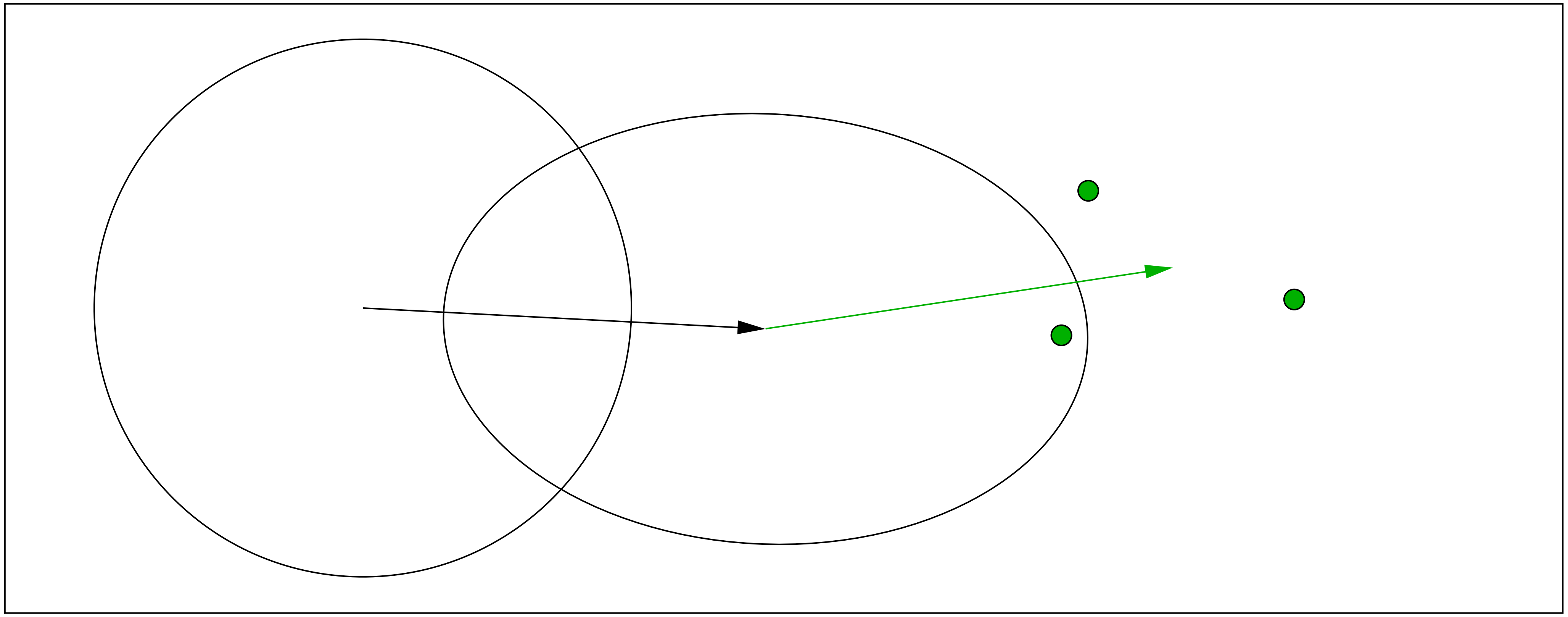


new distribution (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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



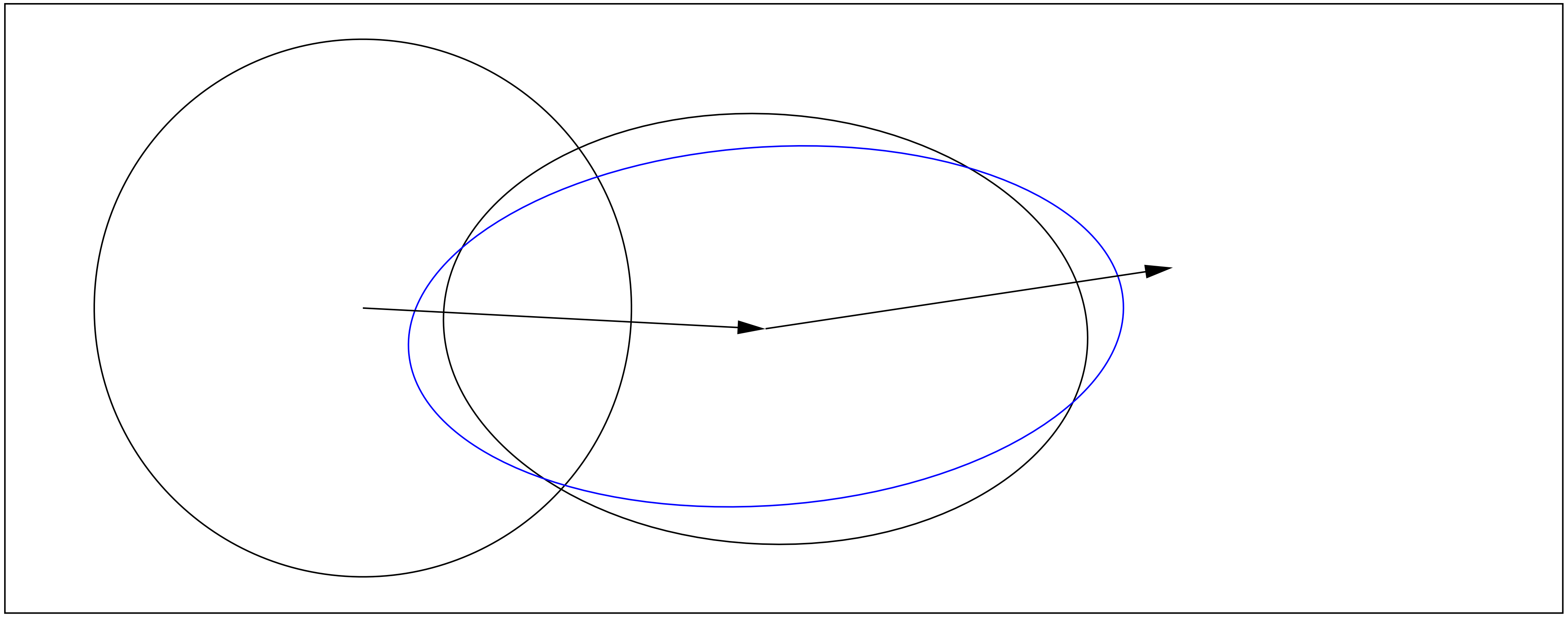
movement of the population mean $\boldsymbol{m}$

. . . equations

Covariance Matrix Adaptation

Rank-One Update

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



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

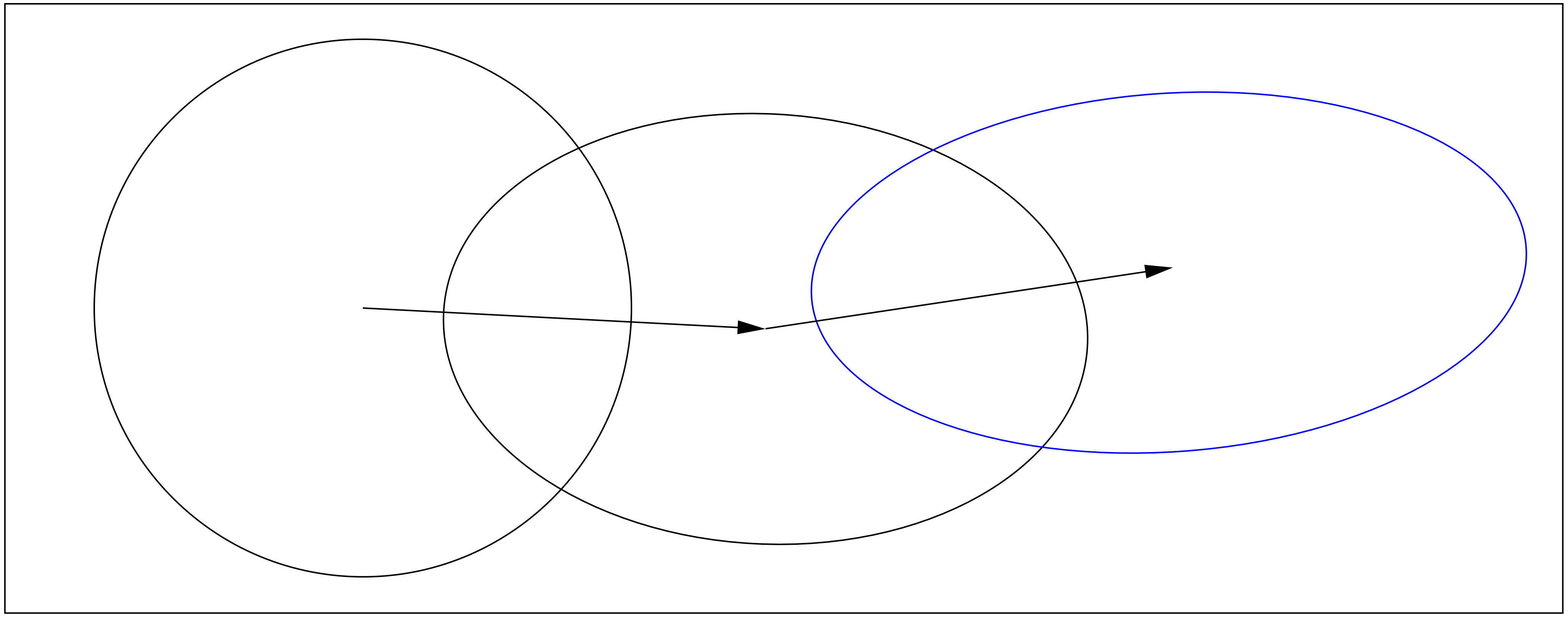
$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$

... equations

Covariance Matrix Adaptation

Rank-One Update

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



new distribution,

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

the ruling principle: the adaptation increases the likelihood of successful steps, $\boldsymbol{y}_w$, to appear again

another viewpoint: the adaptation follows a natural gradient approximation of the expected fitness

Covariance Matrix Adaptation

Rank-One Update

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

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

The rank-one update has been found independently in several domains[6] [7] [8] [9]

[6] Kjellström&Taxén 1981. Stochastic Optimization in System Design, IEEE TCS

[7] Hansen&Ostermeier 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, ICEC

[8] Ljung 1999. System Identification: Theory for the User

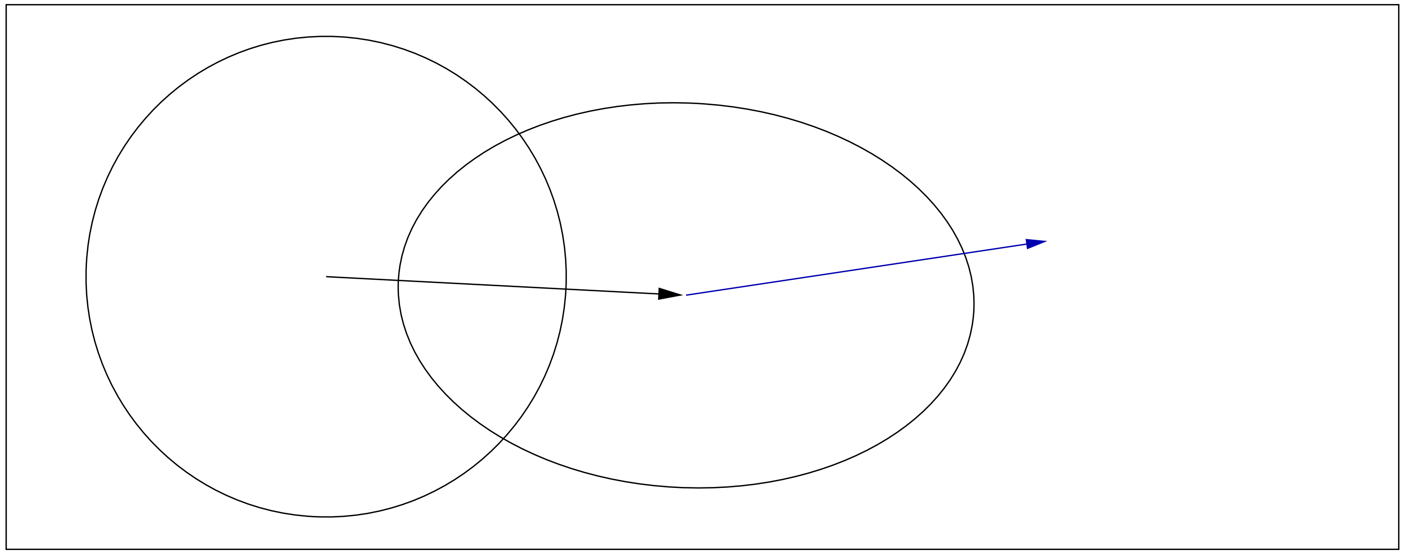
[9] Haario et al 2001. An adaptive Metropolis algorithm, JSTOR

Cumulation

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

Utilizing the Evolution Path

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



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

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

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

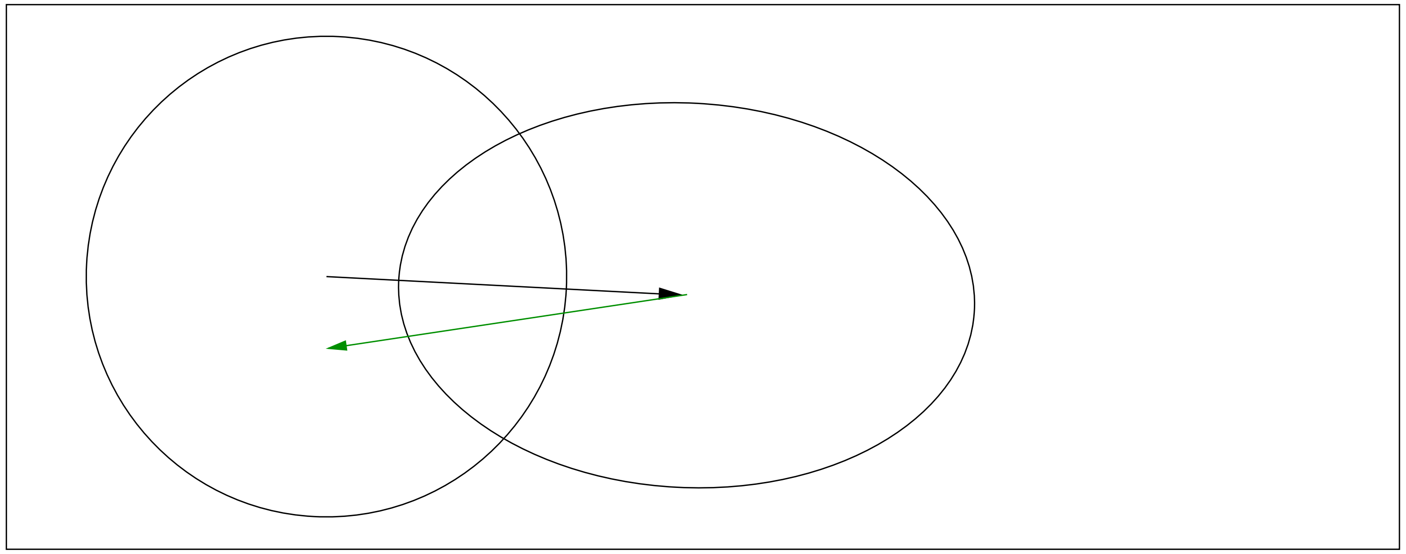
where $\mu_w = \frac{1}{\sum {w_i}^2}$, $c_{\mathrm{cov}} \ll c_{\mathrm{c}} \ll 1$ such that $1/c_{\mathrm{c}}$ is the "backward time horizon".

Cumulation

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

Utilizing the Evolution Path

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



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

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

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

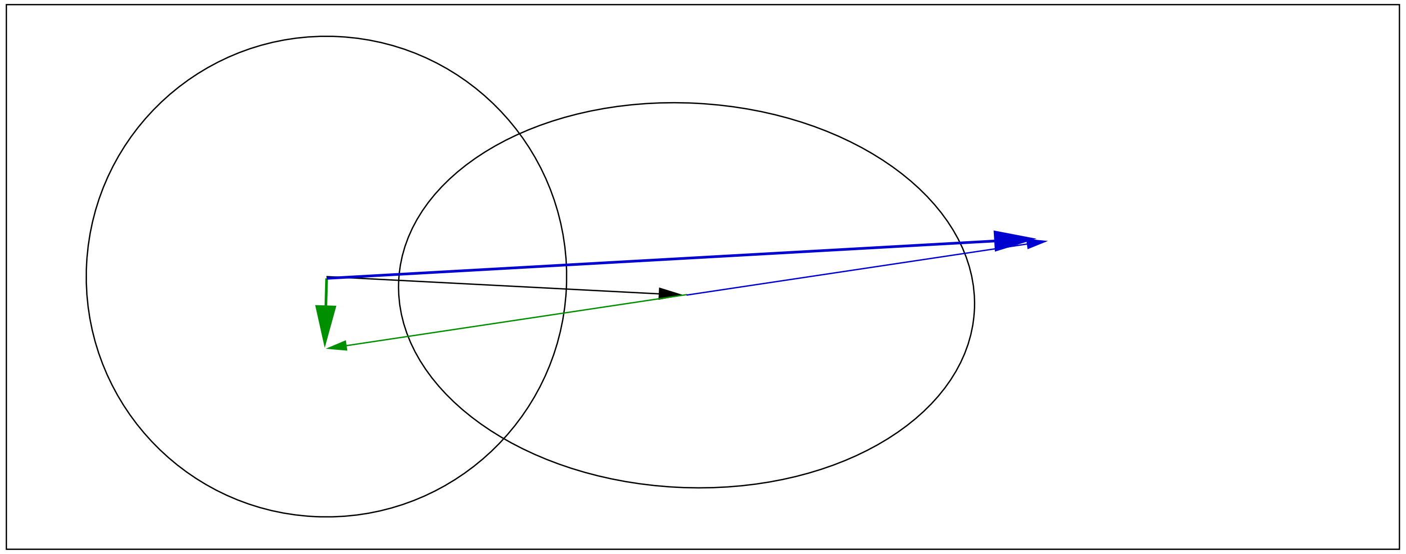
where $\mu_w = \frac{1}{\sum w_i^2}$, $c_{\text{cov}} \ll c_c \ll 1$ such that $1/c_c$ is the "backward time horizon".

Cumulation

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

Utilizing the Evolution Path

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



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

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

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

where $\mu_w = \frac{1}{\sum {w_i}^2}$, $c_{\mathrm{cov}} \ll c_{\mathbf{c}} \ll 1$ such that $1/c_{\mathbf{c}}$ is the "backward time horizon".

Rank-μ Update

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

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

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

computes a weighted mean of the outer products of the best μ steps and has rank $\min(\mu, n)$ with probability one.

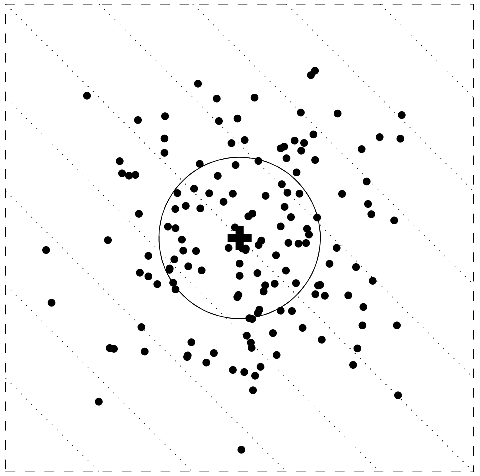
with $\mu = \lambda$ weights can be negative [10]

The rank-μ update then reads

$$\mathbf{C} \leftarrow (1 - c_{\mathrm{cov}})\, \mathbf{C} + c_{\mathrm{cov}}\, \mathbf{C}_\mu$$

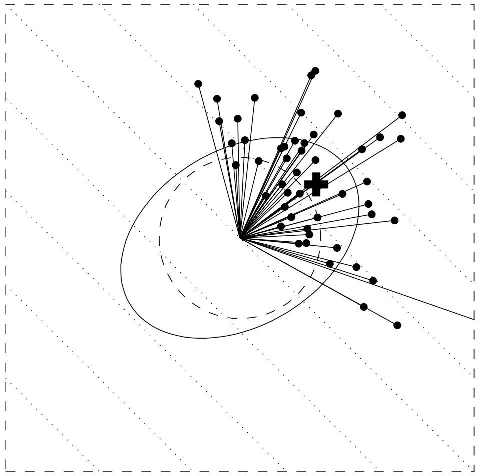
where $c_{\mathrm{cov}} \approx \mu_w / n^2$ and $c_{\mathrm{cov}} \leq 1$.

[10] Jastrebski and Arnold (2006). Improving evolution strategies through active covariance matrix adaptation. CEC.



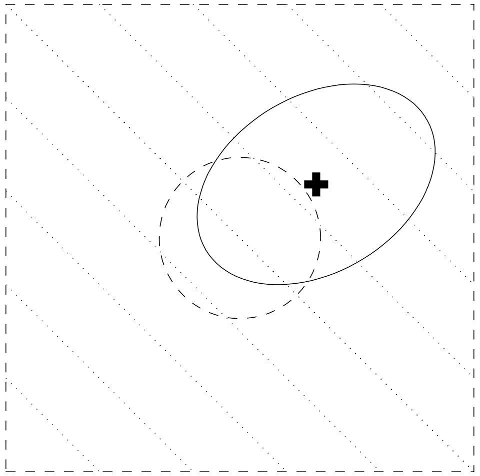
$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\, \boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$

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



$$\mathbf{C}_\mu = \frac{1}{\mu} \sum \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^{\mathrm{T}}$$
$$\mathbf{C} \leftarrow (1-1) \times \mathbf{C} + 1 \times \mathbf{C}_\mu$$

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



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

new distribution

The rank-μ update

- increases the possible learning rate in large populations
  roughly from $2/n^2$ to μ_w/n^2
- can reduce the number of necessary generations roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ [12]
  given $\mu_w \propto \lambda \propto n$

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 reduces the number of necessary function evaluations to learn straight ridges from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

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

. . . all equations

[12] 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: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ (problem dependent)

Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,

Set: $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \approx 0.3\,\lambda$

While not terminate

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

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

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

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

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

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

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

Strategy Internal Parameters

- related to selection and recombination
  - λ, offspring number, new solutions sampled, population size
  - μ, parent number, solutions involved in updates of $\boldsymbol{m}$, $\mathbf{C}$, and σ
  - $w_{i=1,\ldots,\mu}$, recombination weights
- related to $\mathbf{C}$-update
  - $c_{\mathbf{c}}$, decay rate for the evolution path
  - c_1, learning rate for rank-one update of $\mathbf{C}$
  - c_μ, learning rate for rank-μ update of $\mathbf{C}$
- related to σ-update
  - c_σ, decay rate of the evolution path
  - d_σ, damping for σ-change

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

Useful: restarts with increasing population size (IPOP)

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Experimentum Crucis (0)

What did we want to achieve?

- reduce any convex-quadratic function

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

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

to the sphere model

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

without use of derivatives

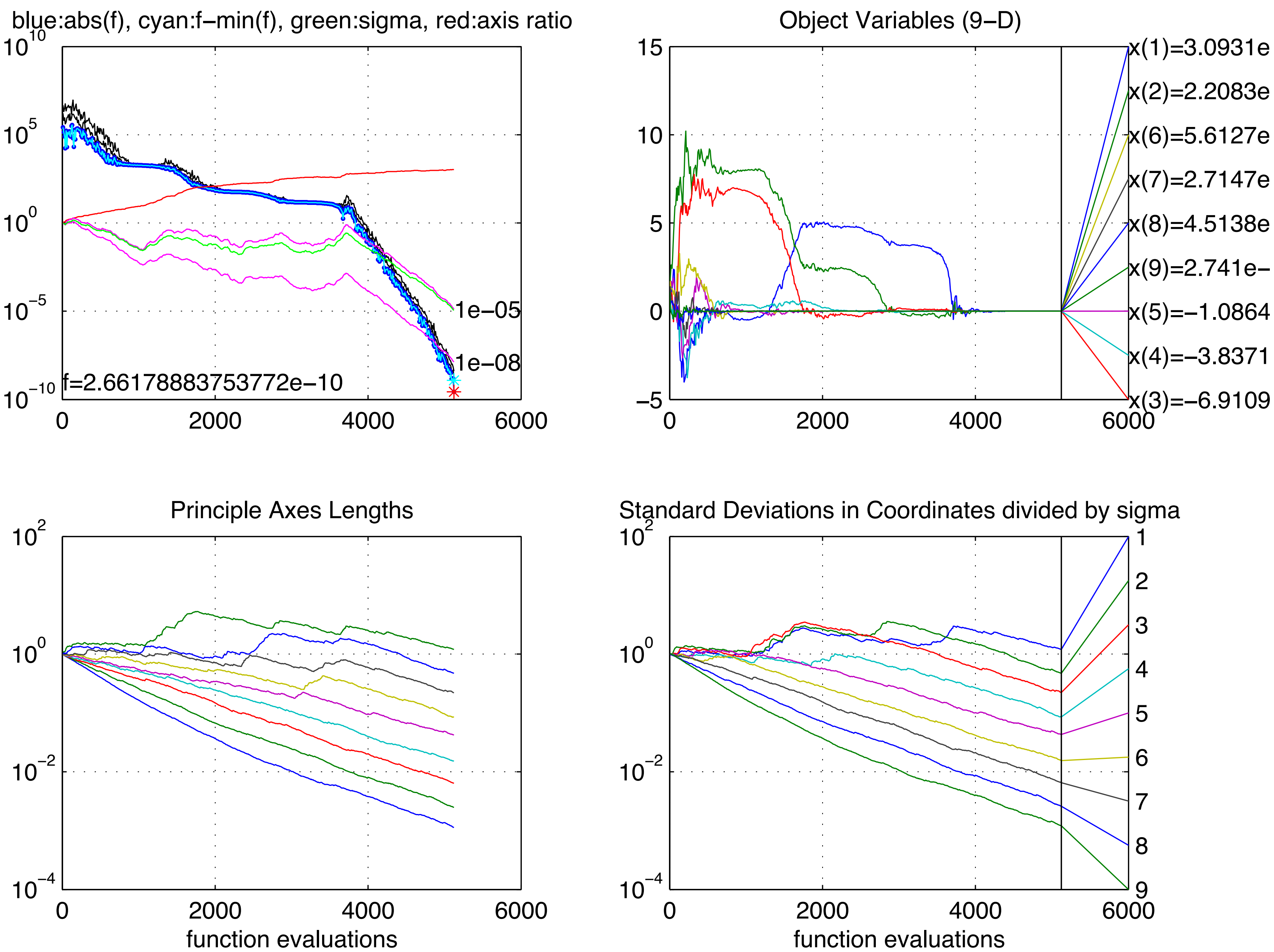
- lines of equal density align with lines of equal fitness

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

in a stochastic sense

Experimentum Crucis (1)

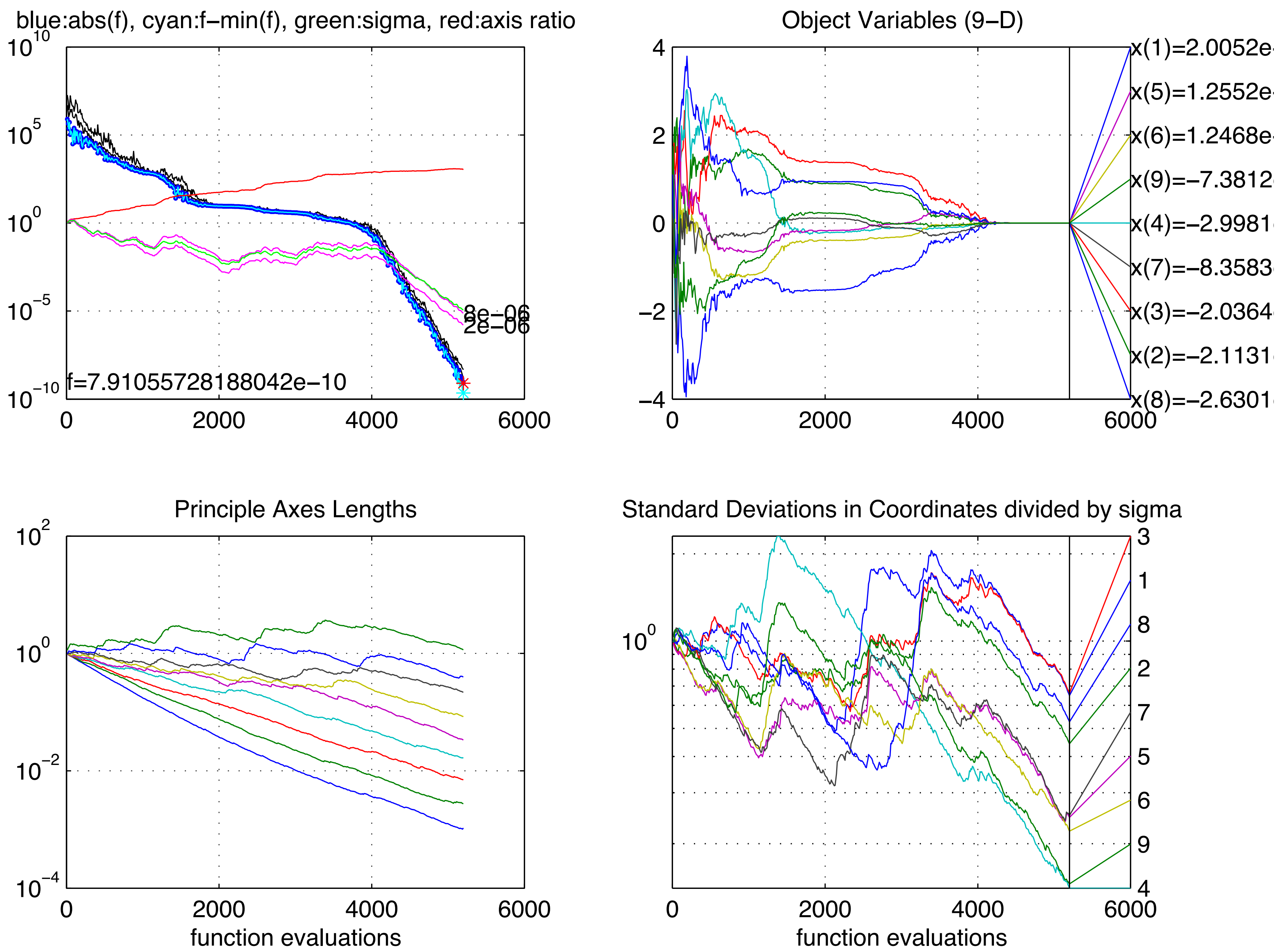
f convex quadratic, separable



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

Experimentum Crucis (2)

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



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

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

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

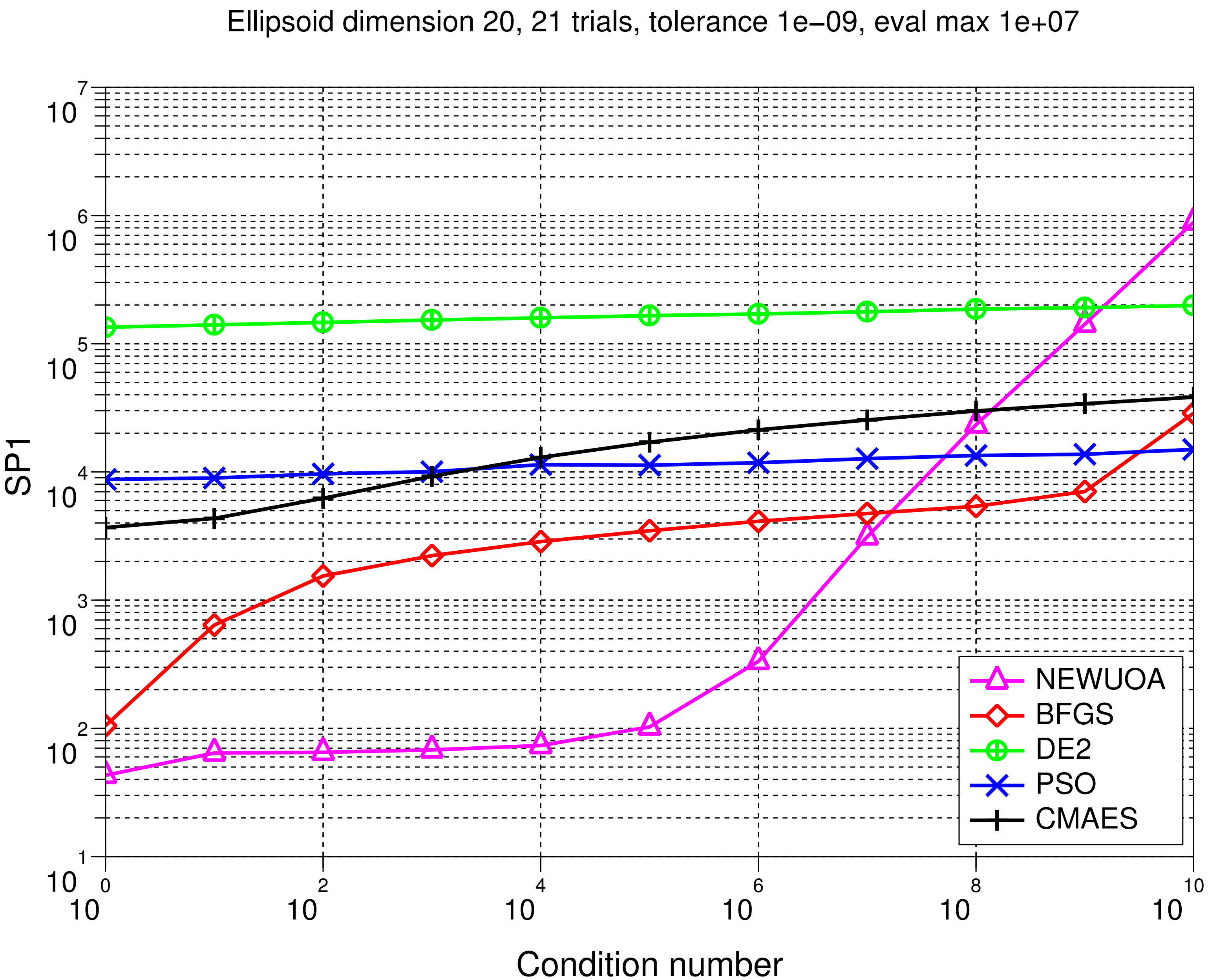
Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Comparison to BFGS, NEWUOA, PSO and DE

f convex quadratic, separable with varying condition number α



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

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

$\boldsymbol{H}$ diagonal

g identity (for BFGS and NEWUOA)

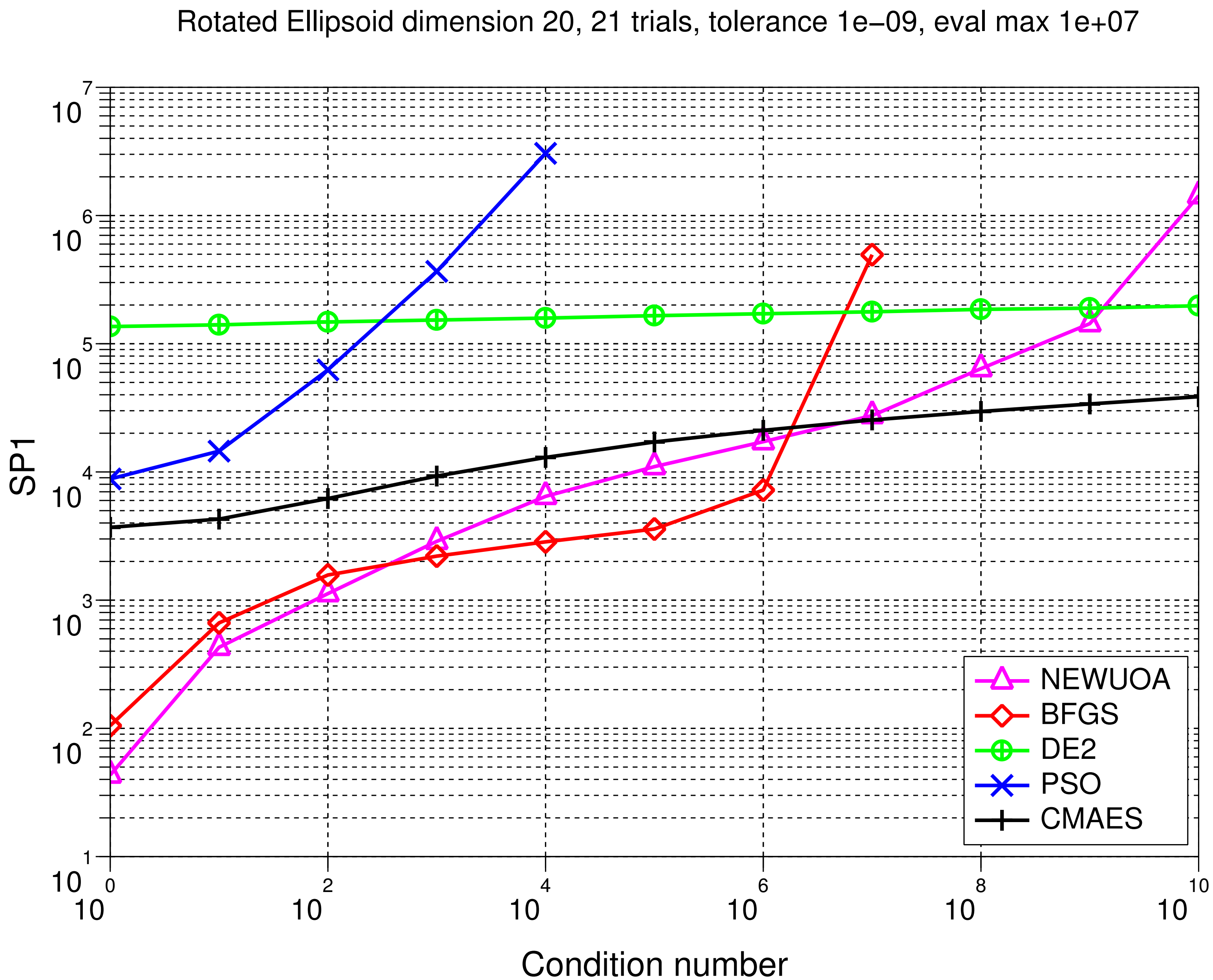
g any order-preserving = strictly increasing function (for all other)

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

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

Comparison to BFGS, NEWUOA, PSO and DE

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



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

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

$\boldsymbol{H}$ full

g identity (for BFGS and NEWUOA)

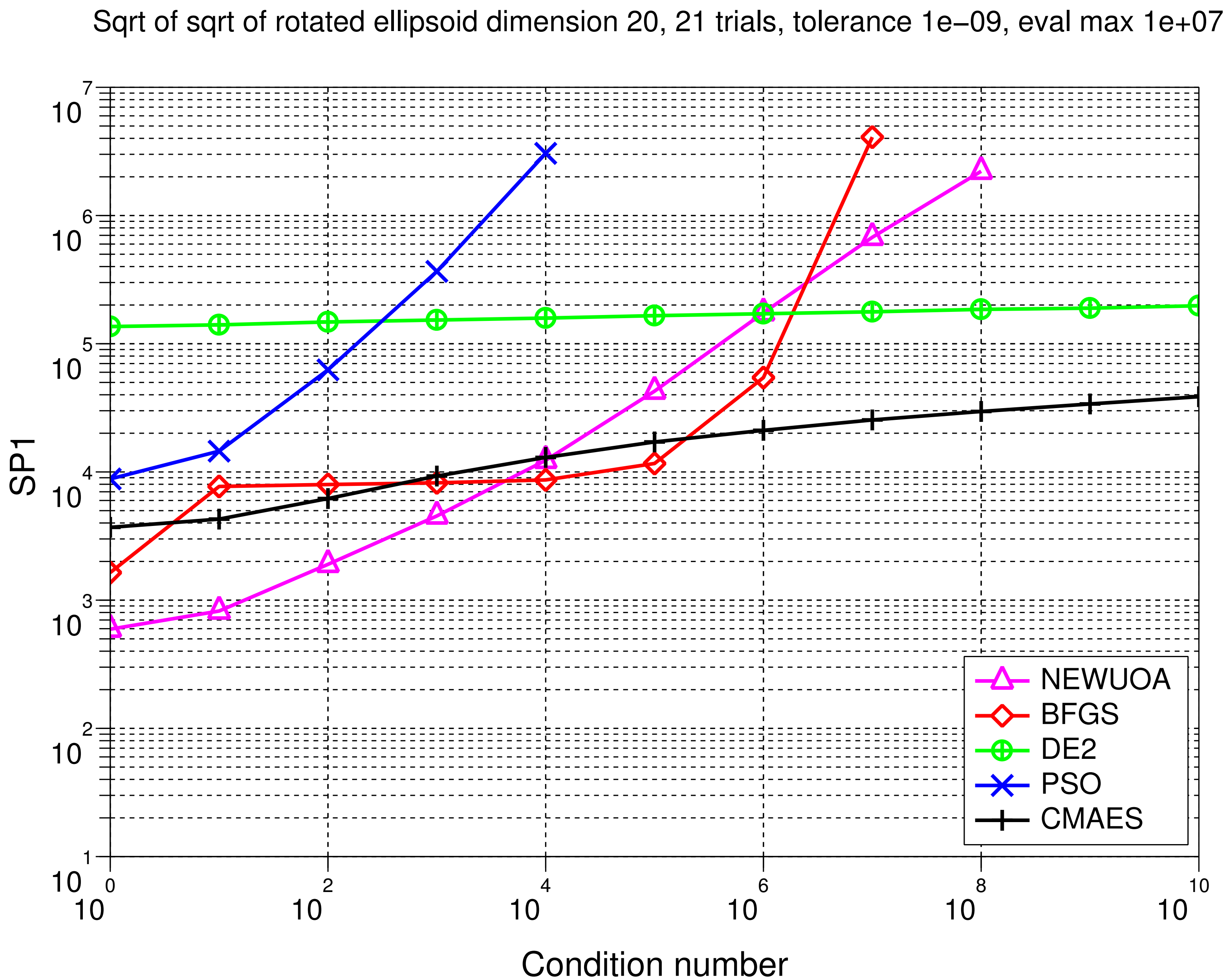
g any order-preserving = strictly increasing function (for all other)

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

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

Comparison to BFGS, NEWUOA, PSO and DE

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



BFGS (Broyden et al 1970)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

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

$\boldsymbol{H}$ full

$g : x \mapsto x^{1/4}$ (for BFGS and NEWUOA)

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

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

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

Overview

General context

What makes an optimization problem difficult?

Insights into CMA-ES

- Adaptation of the mean vector
- Adaptation of the step-size
- Adaptation of the covariance matrix

Variable metric illustration - learning inverse Hessian

Local versus global search - comparisons with BFGS / NEWUOA

Theoretical aspects

- Invariance
- Connexion with gradient optimization on manifolds (information geometry)

Invariance

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

— Albert Einstein

Empirical performance results

from test functions

from solved real world problems

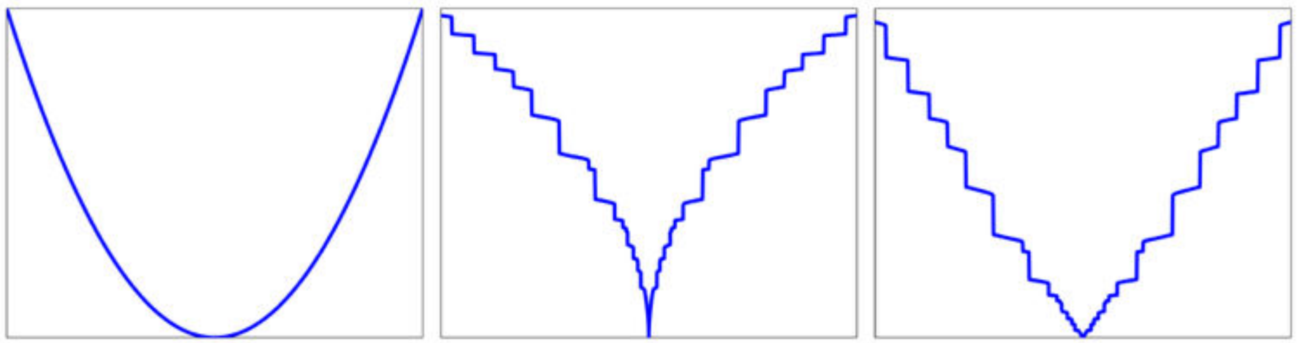
are only useful if they do generalize to other problems

Invariance is a strong non-empirical statement about generalization

generalizing performance from a single function to a whole class of functions

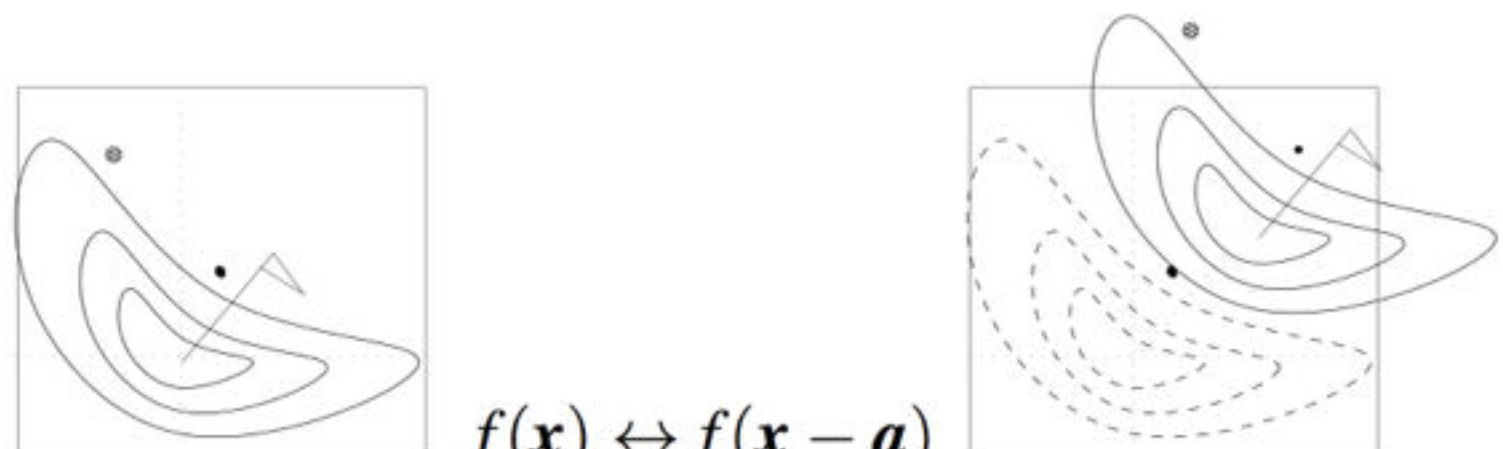
Invariances of CMA-ES

Invariance under monotonically increasing functions



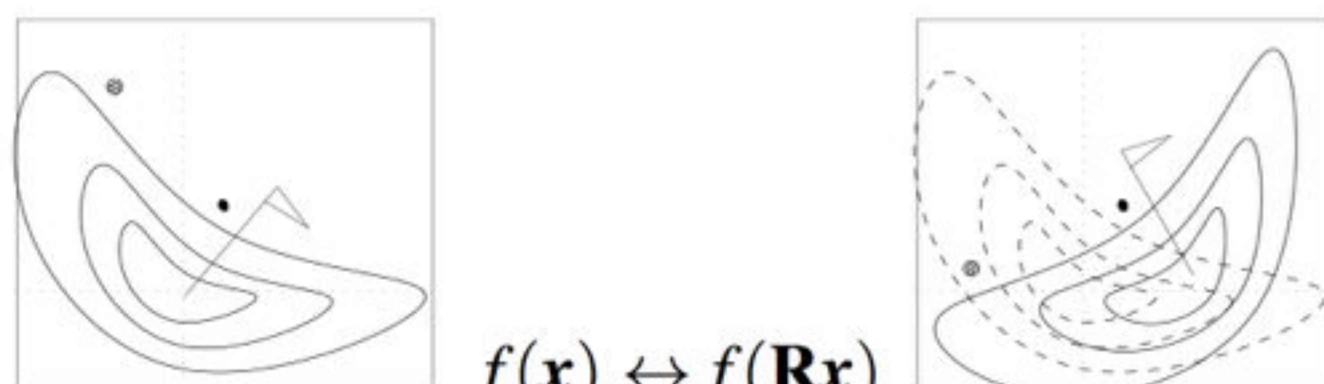
comparison-based

Translation invariance



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

Rotational invariance



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

Identical performance

Invariances of CMA-ES (cont.)

Scale invariance

Identical performance on $f(\mathbf{x}) \leftrightarrow f\left(\alpha\mathbf{x}\right), \alpha > 0$

Affine invariance

Identical performance on $f(\mathbf{x}) \leftrightarrow f\left(A\mathbf{x} + b\right), A \in \mathrm{GL}(n, \mathbb{R}), b \in \mathbb{R}^n$

Affine invariance $\Rightarrow$ scale invariance
rotational invariance
translation invariance

Invariance for several BB optimizers

	comparison-based	translation	scale	rotation	affine
BFGS		X	X	X	X
NEWUOA		X	X		
Nelder-Mead	X	X	X	X	X
CMA-ES	X	X	X	X	X
PSO	X	X	X		

Connexion with Optimization on Manifolds
Information Geometric Optimization

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

Initialize distribution parameters θ, set population size λ
While not terminate

1. Sample distribution $p_\theta(x) : x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on f
3. Update parameters $\theta \leftarrow F(\theta, x_1, \ldots, x_\lambda, f(x_1), \ldots, f(x_\lambda))$

formal way to write part of CMA-ES θ-update as a gradient step on statistical manifold formed by the family of probability distribution p_θ

Connexion with Optimization on Manifolds

Information Geometric Optimization

- Transform original problem into optimization problem on the statistical manifold Θ where p_θ lives

$$\text{Minimize } J(\theta) = \int f(x) p_\theta(x) dx$$

not invariant to mont. transformation of f

Wiestra et al. Natural Evolution Strategies, CEC 2008
Sun et al. Efficient natural evolution strategies GECCO 2009
Glasmachers et al. Exponential NES GECCO 2010

$$\text{Maximize } J_{\theta_t}(\theta) = \int w(P_{\theta_t}[y : f(y) \leq f(x)]) p_\theta(x) dx$$

$w : [0, 1] \mapsto \mathbb{R}$, decreasing weight function

Ollivier et al. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles, arXiv

Connexion with Optimization on Manifolds

Information Geometric Optimization

☆ Perform a natural gradient step on Θ

gradient taken w.r.t. Fisher Information metric $I_{ij} = \int \frac{\partial \log p_\theta(x)}{\partial \theta_i} \frac{\partial \log p_\theta(x)}{\partial \theta_j} p_\theta(x) dx$

$$\tilde{\nabla}_\theta = I^{-1} \frac{\partial}{\partial \theta}$$

$$\begin{aligned} \theta_{t+\delta t} &= \theta_t + \delta t \, \tilde{\nabla} J_{\theta_t}(\theta)|_{\theta=\theta_t} \\ &= \theta_t + \delta t \int w(p_{\theta_t}[y : f(y) \leq f(x)]) \tilde{\nabla}_\theta \ln p_\theta(x) \, |_{\theta=\theta_t} p_{\theta_t}(x) dx \end{aligned}$$

Connexion with Optimization on Manifolds

Information Geometric Optimization

☆ Monte Carlo approximation of the integral

$$\theta_{t+\delta t} = \theta_t + \delta t \int w(p_{\theta_t}[y : f(y) \leq f(x)]) \tilde{\nabla}_\theta \ln p_\theta(x) \mid_{\theta=\theta_t} p_{\theta_t}(x) dx$$

Sample $X_i \sim p_{\theta_t}(x), i = 1, \ldots \lambda$

$$\theta_{t+1} = \theta_t + \delta t \frac{1}{\lambda} \sum_{i=1}^{\lambda} w_{rk(X_i)} \tilde{\nabla}_\theta \ln p_\theta(X_i)$$

For p_θ family of Gaussian distribution $\theta = (\mathbf{m}, \mathbf{C})$

sample $\mathrm{x}_i \sim \mathcal{N}(m_t, C_t), \quad i = 1 \ldots, \lambda$

$$m_{t+\delta t} = m_t + \delta t \tfrac{1}{\lambda} \textstyle\sum_{i=1}^{\lambda} w(rk(\mathrm{x}_i))(\mathrm{x}_i - m_t)$$

$$C_{t+\delta t} = C_t + \delta t \tfrac{1}{\lambda} \textstyle\sum_{i=1}^{\lambda} w(rk(\mathrm{x}_i))(x_i - m_t)(x_i - m_t)^T - C_t$$

CMA-ES with rank-mu update

Akimoto et al. *Bidirectional relation between CMA evolution strategies and natural evolution strategies*, 2010 PPSN XI

Limitations

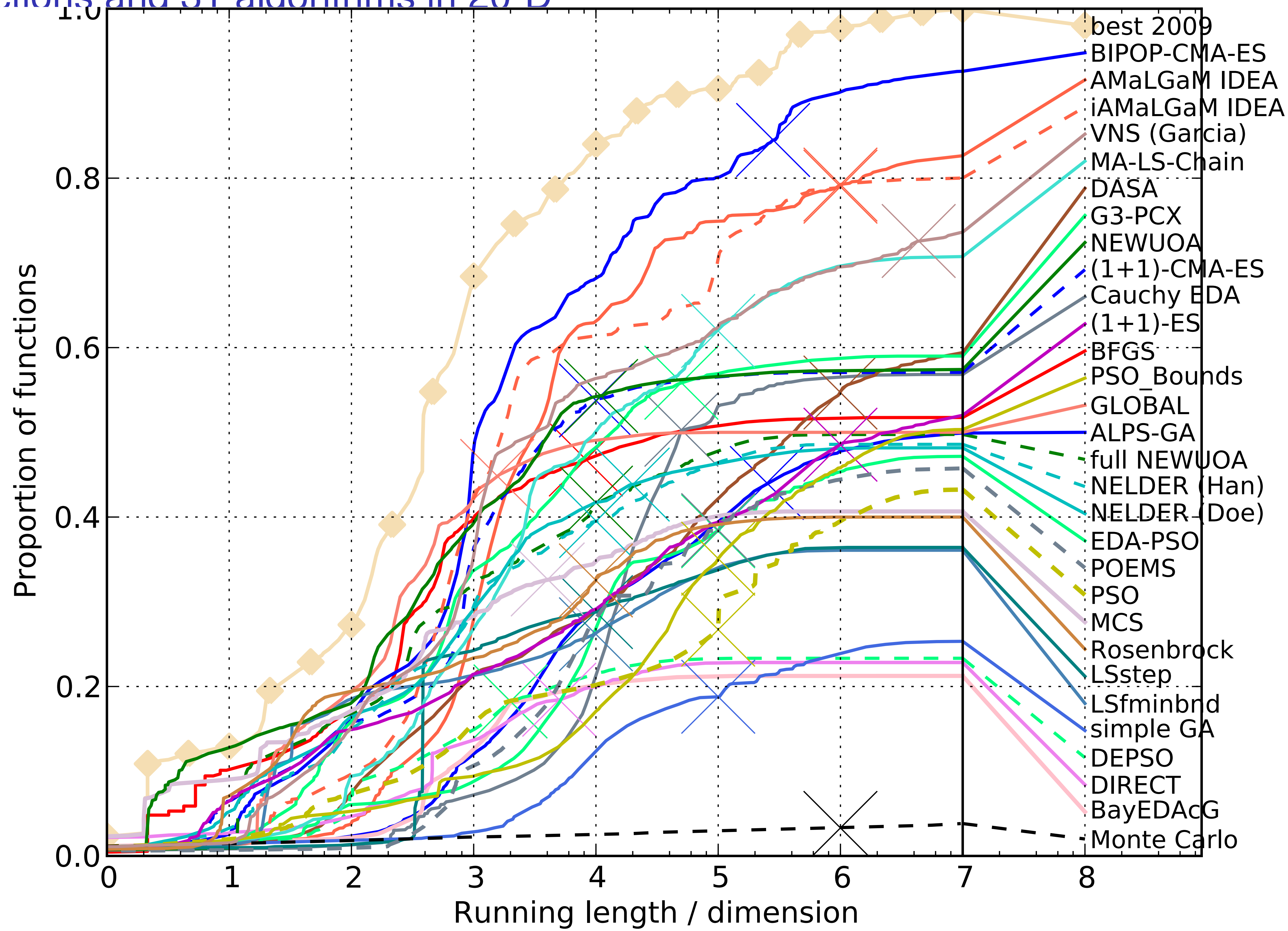
of CMA Evolution Strategies

- internal CPU-time: $10^{-8}n^2$ seconds per function evaluation on a 2GHz PC, tweaks are available
  1 000 000 f-evaluations in 100-D take 100 seconds *internal* CPU-time
- better methods are presumably available in case of
  - partly separable problems
  - specific problems, for example with cheap gradients
    specific methods
  - small dimension ($n \ll 10$)
    for example Nelder-Mead
  - small running times (number of f-evaluations $< 100n$)
    model-based methods

Thank you!

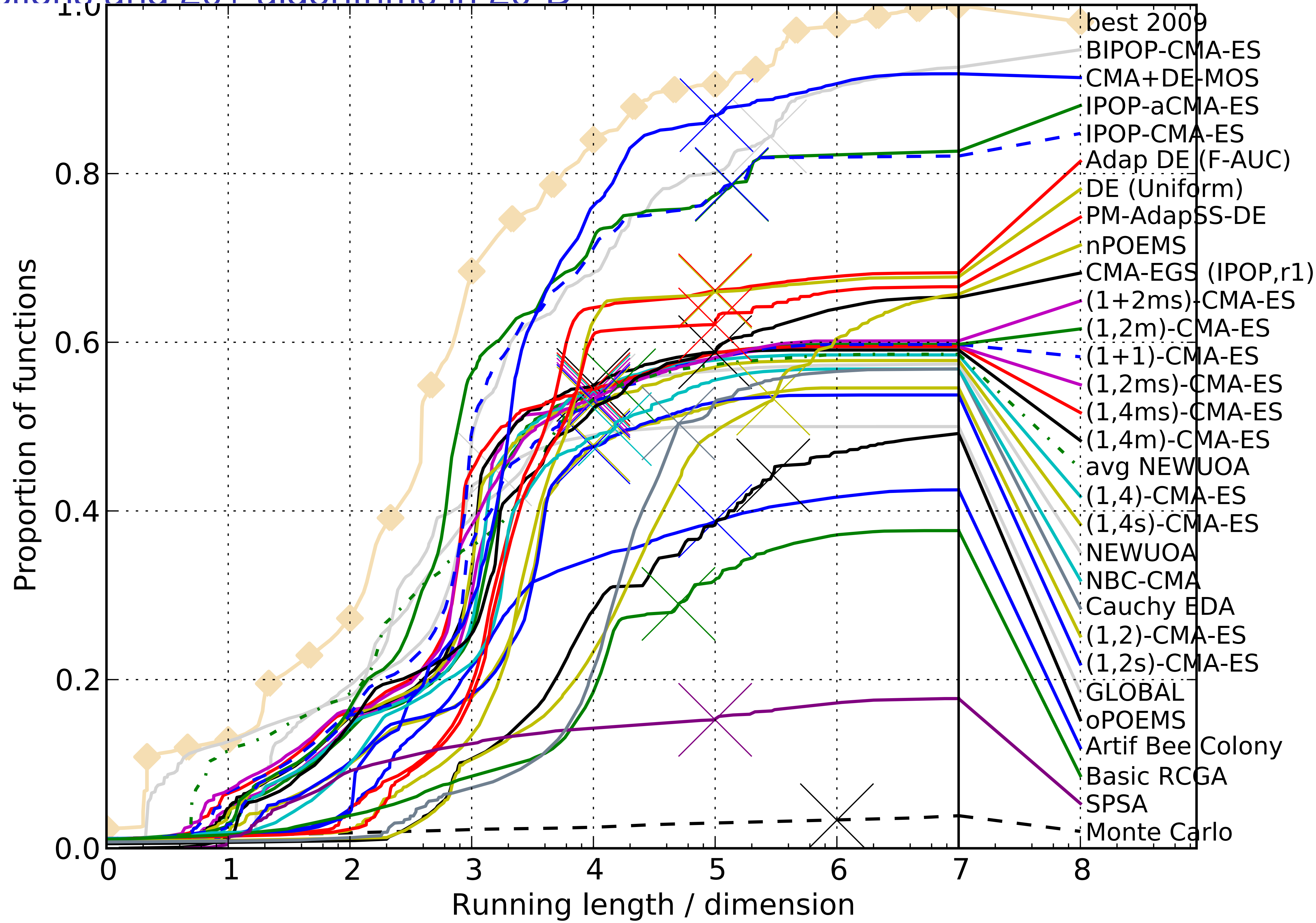
Comparison during BBOB at GECCO 2009

24 functions and 31 algorithms in 20-D



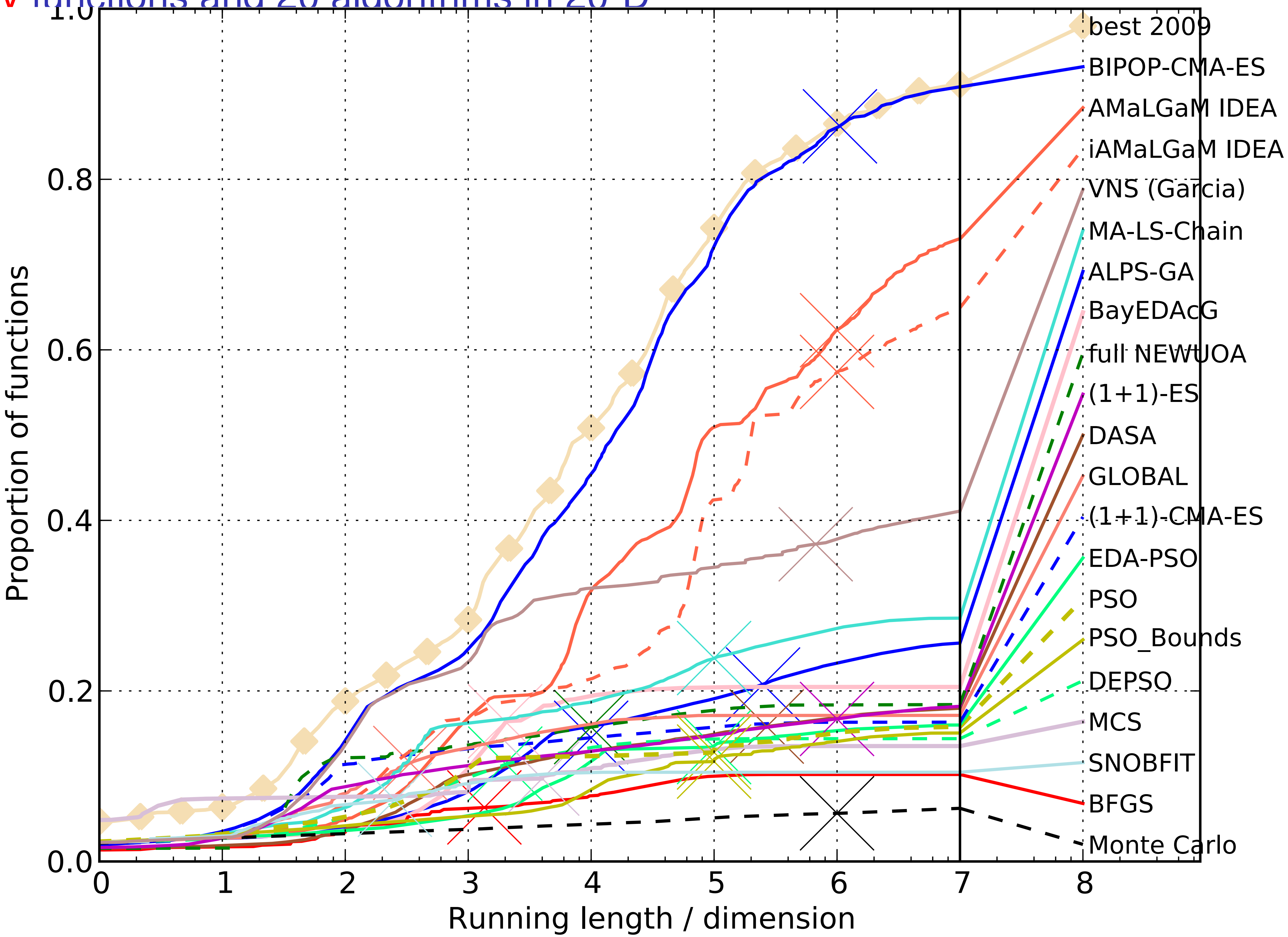
Comparison during BBOB at GECCO 2010

24 functions and 20+ algorithms in 20-D



Comparison during BBOB at GECCO 2009

30 noisy functions and 20 algorithms in 20-D



Comparison during BBOB at GECCO 2010

30 noisy functions and 10+ algorithms in 20-D