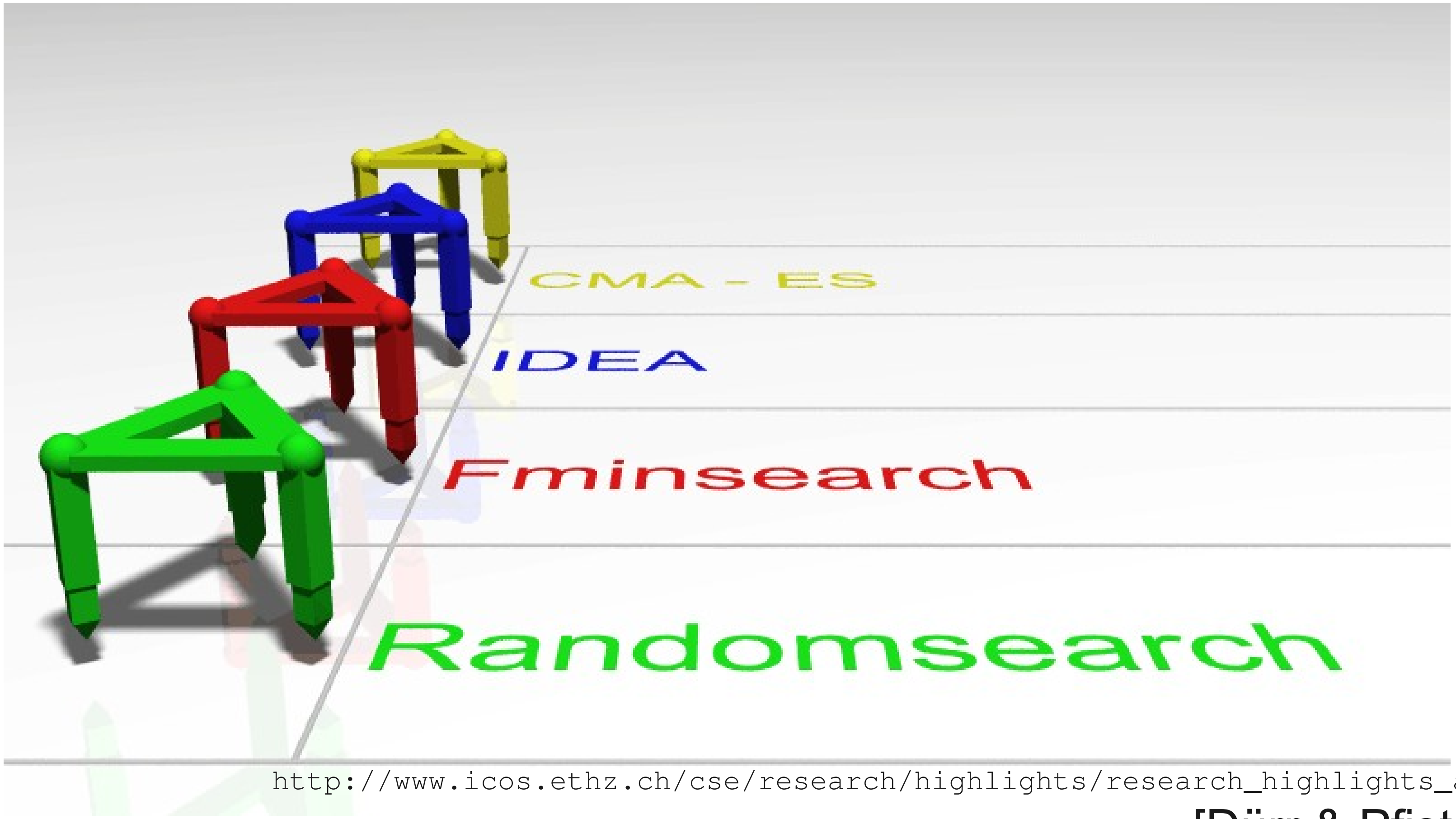
SIMINOLE Task 3: Simulation-based stochastic optimization

Nikolaus Hansen (manager)
INRIA Team TAO (Apprentissage & Optimisation)

...please ask questions...

Simulation-based optimization: an example

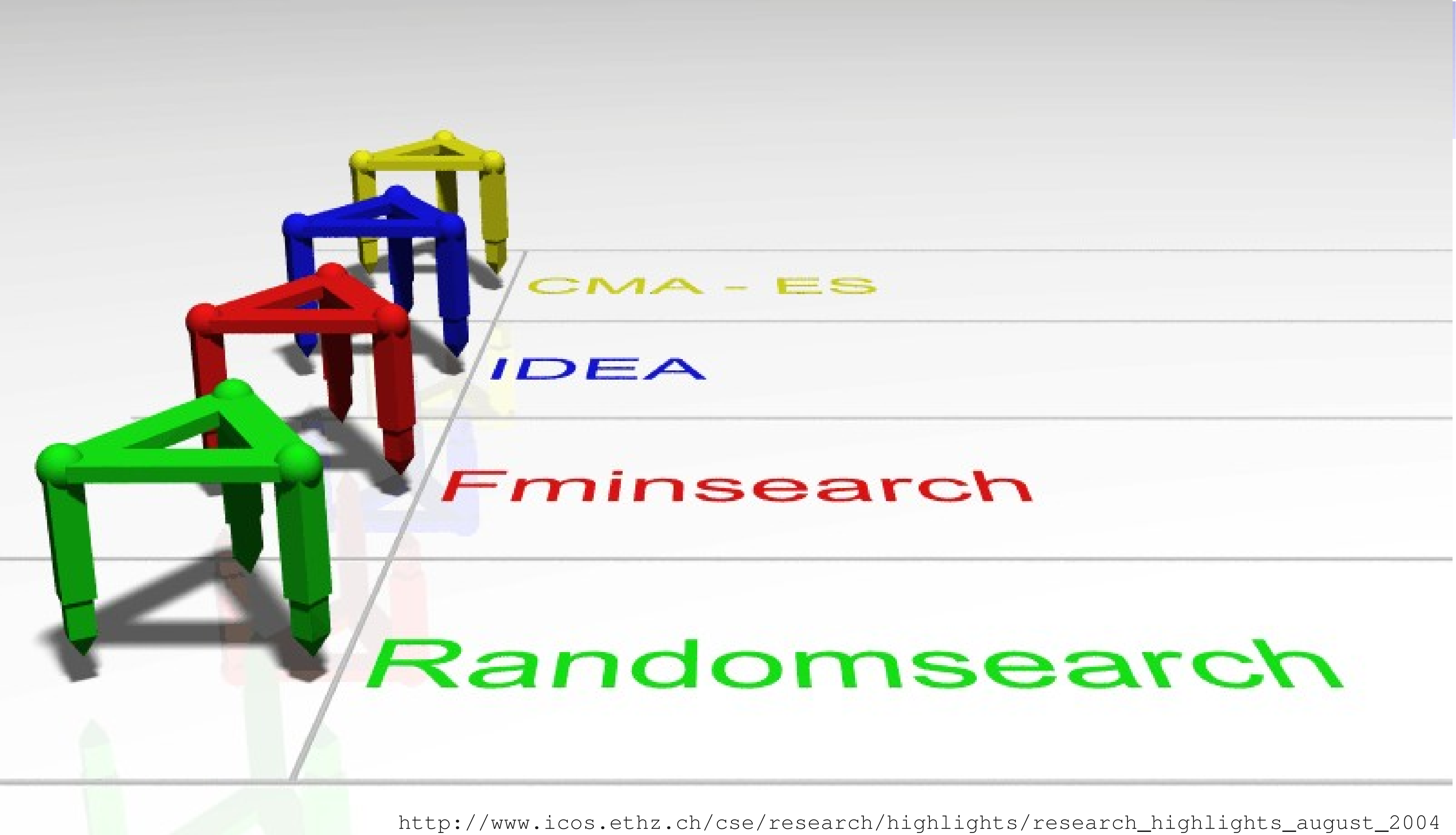
Optimization of walking gaits



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]



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]

Black-Box Optimization (Search)

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

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

where f is simulated (depicted as a black-box)

and in particular

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

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

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

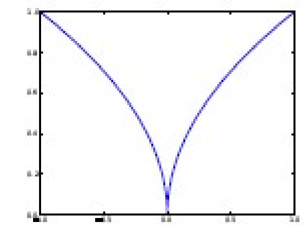
  dependencies between the objective variables

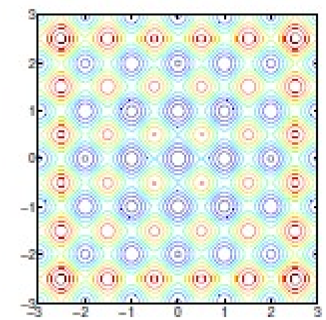
- ill-conditioning

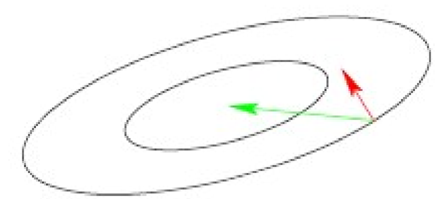
  widely varying sensitivity

- ruggedness

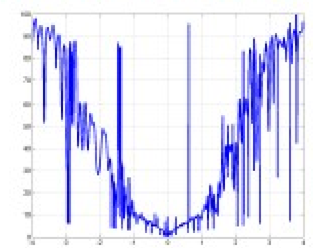
  non-smooth, discontinuous, multimodal, and/or noisy function







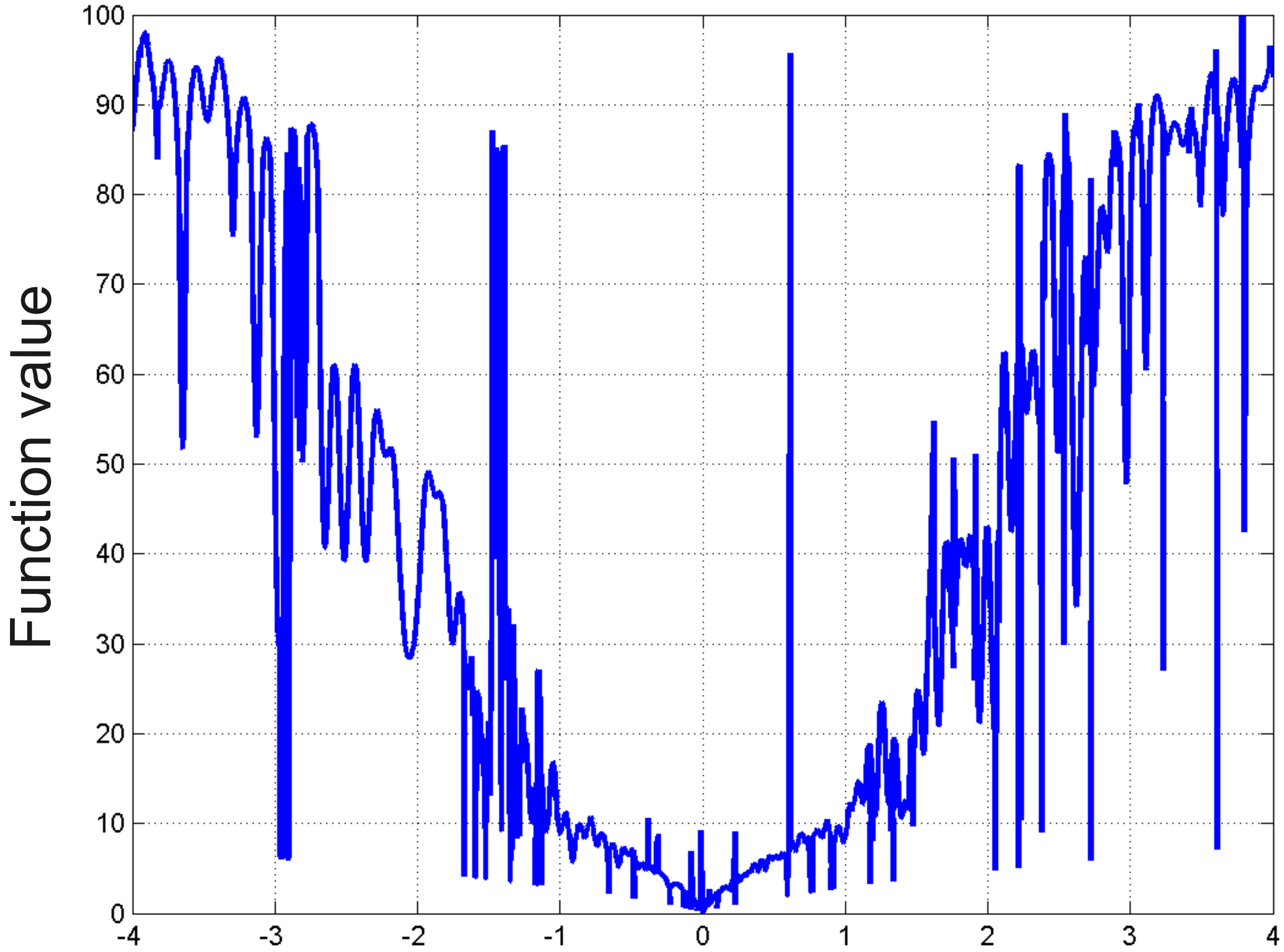
gradient direction Newton direction



in any case the objective function must be highly regular

Rugged landscape

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



Black-Box Optimization 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]

...a principled view point...

Principled Stochastic Optimization

Consider a sample distribution $P(.|\theta)$ with density p
consider $E(f(x)|\theta)$ to be minimized w.r.t. θ
consider $\nabla_\theta E(f(x)|\theta)$ for updating θ
rather consider $\tilde{\nabla}_\theta E(f(x)|\theta)$, as the natural gradient is independent of the parameterization

$$\begin{aligned}
\tilde{\nabla}_\theta E(f(x)|\theta) &= F_\theta{}^{-1} \nabla_\theta E(f(x)|\theta) \\
&= E(f(x) F_\theta{}^{-1} \nabla_\theta \ln p(x|\theta)) \\
&\approx \frac{1}{\lambda} \sum_{i=1}^{\lambda} f(x_i) F_\theta{}^{-1} \nabla_\theta \ln p(x_i|\theta)
\end{aligned}$$

where F_θ is the Fisher information matrix and $x_i \sim p(.|\theta)$ for $i = 1 \ldots \lambda$

suggests a stochastic steepest descend

using the maximum entropy distribution for p, where $F_\theta{}^{-1} \nabla_\theta \ln p(x_i|\theta)$ is known, and two additional trick/design principle leads to…

A Natural Evolution Strategy

Natural Evolution Strategy = CMA-ES – step-size control – cumulation

Input: $m \in \mathbb{R}^n$, $\lambda \in \{2, 3, 4, \dots\}$

Set $c_\mu \approx \mu_w / n^2$, set $w_{i=1,\dots,\lambda} > 0$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \approx 0.3\,\lambda$

Initialize $\mathbf{C} = \mathbf{I}$,

While not *terminate*

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

$$m \leftarrow m + \sum_{i=1}^{\mu} w_i (\mathbf{x}_{i:\lambda} - m), \quad f(\mathbf{x}_{1:\lambda}) \le f(\mathbf{x}_{2:\lambda}) \dots \qquad \text{update mean}$$

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

using fixed weights w_i instead of the function values $f(x_i)$ and
using different learning rates (step-sizes) for m and $\mathbf{C}$
adding a few more tricks and design principles
leads to. . .

Covariance Matrix Adaptation Evolution Strategy

CMA-ES = natural gradient descent + cumulation + step-size control

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

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

Initialize $\mathbf{C} = \mathbf{I}$, and $\mathbf{p}_c = \mathbf{0}$, $\mathbf{p}_\sigma = \mathbf{0}$

While not *terminate*

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

$$m \leftarrow \sum_{i=1}^{\mu} w_i\,\mathbf{x}_{i:\lambda} = m + \sigma\,\mathbf{y}_w, \quad f(\mathbf{x}_{1:\lambda}) \leq f(\mathbf{x}_{2:\lambda}) \ldots \qquad \text{update mean}$$

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

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

$$\mathbf{p}_c \leftarrow (1 - c_c)\,\mathbf{p}_c + \mathbb{1}_{[0,1.5]}\left(\frac{\|\mathbf{p}_\sigma\|}{\sqrt{n}}\right)\sqrt{1-(1-c_c)^2}\sqrt{\mu_w}\,\mathbf{y}_w \qquad \text{path for } \mathbf{C}$$

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

Covariance Matrix Adaptation Evolution Strategy

CMA-ES = natural gradient descent + cumulation + step-size control

While not *terminate*

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

$$m \leftarrow \sum_{i=1}^{\mu} w_i\,\mathbf{x}_{i:\lambda} = m + \underbrace{\sigma\mathbf{y}_w}_{\text{iterate displacement}} = m + \sigma\sum_{i=1}^{\mu} w_i\,\mathbf{y}_{i:\lambda} \qquad \text{update mean}$$

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

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

$$\mathbf{p}_c \leftarrow \underbrace{(1 - c_c)}_{\text{discount factor}}\,\mathbf{p}_c + \underbrace{\mathbb{1}_{[0,1.5]}\left(\frac{\|\mathbf{p}_\sigma\|}{\sqrt{n}}\right)}_{\text{stall}}\sqrt{1 - (1 - c_c)^2}\,\underbrace{\sqrt{\mu_w}\,\mathbf{y}_w}_{\text{under neutral selection } \mathcal{N}(\mathbf{0},\mathbf{C})} \qquad \text{path for } \mathbf{C}$$

$$\mathbf{C} \leftarrow \underbrace{(1 - c_1 - c_\mu)}_{\text{discount factor}}\,\mathbf{C} + c_1\,\underbrace{\mathbf{p}_c\mathbf{p}_c^{\mathrm{T}}}_{\text{rank one}} + c_\mu\underbrace{\textstyle\sum_{i=1}^{\mu} w_i\,\mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^{\mathrm{T}}}_{\text{rank } \mu} \qquad \text{update } \mathbf{C}$$

Known Issues

- Multi-funnel landscapes often pose difficulties
- Scaling with the search space dimension is typically sub-quadratic

  in large dimensions linear scaling is desirable
- How to evaluate this (any such kind of) algorithm?
- Is there a deeper principled reasoning for the additionally introduced tricks?

The Plan: Task 3.1

Addressing complex, e.g. multi-funnel landscapes by

- Coupling mixtures of Gaussians with the CMA-ES update principles (natural gradient descent + cumulation + step-size control)
  - …
  - ...
- Derive “CMA”-ES variants which can learn more complex (non-linear) dependencies
  - Based on non-linear projections & PCA
  - Based on independent component analysis
  - ...

The Plan: Task 3.2

Addressing large-scale problems by

- Deriving “simplified CMA”-ES variants with linear scaling
  - Linear in black-box (function) evaluations
  - Linear in internal CPU-time
  - Linear in memory
  - Objective: still solve comparatively complex (e.g. highly non-separable multimodal) functions

The Plan: Task 3.3

Performance evaluation of black-box optimization algorithms

- COCO: a platform for COmparing Continuous Optimisers has been started in 2009
  - Characterization of simulation-based optimization problems → benchmark function set
  - Performance indicators
  - Performance data presentation and interpretation

The Plan: Task 3.4

- Can we find a principled motivation for
  - different learning rates in the natural gradient descend?
  - cumulation?
  - step-size control?

(more) question?

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