# A linear time natural gradient algorithm for black-box optimization in high dimension

Youhei Akimoto[1] Anne Auger[2] Nikolaus Hansen[2]

[1]Shinshu University, Japan
[2]INRIA - Research Center Saclay, France

PGMO-COPI'14, Paris-Saclay, France, Oct. 28–31, 2014

# Black-Box Optimization in High Dimension

In Black-Box Optimization,

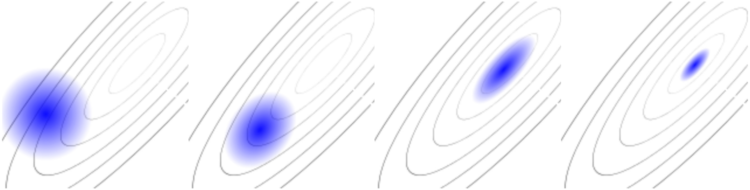
- *no priori knowledge* about the problem; The objective function may be *nonsmooth*, *nonconvex*, *discontinous*, *rugged*.
- $f$-evaluation is computationally expensive; it typically requires a simulation. Therefore, the objective is to minimize the number of $f$-evaluations.
- in high dimension, *the internal time/space complexity of a search algorithm can be the bottleneck*; e.g., ($d$: number of variables)
  - $\mathcal{O}(d)$ for $f$-call
  - $\mathcal{O}(d^2)$ for internal time/space complexity of a search algorithm

Objective: A linear time/space algorithm for black-box optimization in high dimension.

# Covariance Matrix Adaptation Evolution Strategy

The CMA-ES[1] is a quasi parameter free and stochastic black-box continuous optimization algorithm.

- Candidates $(x_i^{(t)})_{i=1,\dots,\lambda}$ are drawn from $\mathcal{N}(\mathbf{m}^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)})$.
- Sort $(x_i^{(t)})_{i=1,\dots,\lambda}$ with respect to $f$-value.
- Update $\mathbf{m}^{(t)}$, $\sigma^{(t)}$, and $\mathbf{C}^{(t)}$ using the sorted solutions $(x_{i:\lambda})_{i=1,\dots,\lambda}$ according to two principle: Natural Gradient and Cumulation



[1] Nikolaus Hansen and Anne Auger. "Principled Design of Continuous Stochastic Search: From Theory to Practice". In: *Theory and Principled Methods for the Design of Metaheuristics*. Ed. by Y Borenstein and A Moraglio. Springer, 2013. ISBN: 978-3-642-33205-0. URL: http://hal.inria.fr/hal-00808450/.

# Invariance of the CMA-ES

- Invariant to strictly increasing map $g$ of $f$; $f \sim g \circ f$
- Invariant to translation; $f(x) \sim f(x_0 + x)$
- Invariant to linear transformation; $f(x) \sim f(Ax)$
  - invariant to rotation of search space $x \mapsto Rx$
    $R$: orthogonal
  - invariant to scaling of search space $x \mapsto Dx$
    $D$: diagonal

# Computational Complexity

$\mathcal{O}(\lambda d^2)$ floating point (FP) multiplication + $\mathcal{O}(d^2 + \lambda d)$ FP memory. ($\lambda \in o(d)$)

1. $\sqrt{\mathbf{C}^{(t)}} = \text{MatrixSqrt}(\mathbf{C}^{(t)})$ (perform every $\mathcal{O}(d/\lambda)$ iter.)
2. $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 1, \ldots, \lambda$
3. $x_i = \mathbf{m}^{(t)} + \sigma^{(t)} \sqrt{\mathbf{C}^{(t)}} z_i$

$\mathcal{O}(\lambda d^2)$

4. $(x_{i:\lambda})_{i=1,\ldots,\lambda} = \text{SortW.r.t.}^f((x_i)_{i=1,\ldots,\lambda})$

5. $\mathbf{p}_{\mathbf{C}}^{(t+1)} = (1 - c_c)\mathbf{p}_{\mathbf{C}}^{(t)} + \sqrt{c_c(2 - c_c)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i (x_{i:\lambda} - \mathbf{m}^{(t)})/\sigma^{(t)}$
6. $\mathbf{p}_{\sigma}^{(t+1)} = (1 - c_\sigma)\mathbf{p}_{\sigma}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i z_{i:\lambda}$
7. $\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(t+1)}\|}{\mathbb{E}[\|\mathcal{N}(\mathbf{0},\mathbf{I})\|]} - 1\right)\right)$

$\mathcal{O}(\lambda d)$

8. $\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{m}} l(\theta^{(t)}; x_{i:\lambda})$
9. $\mathbf{C}^{(t+1)} =$ $\mathbf{C}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$

$\mathcal{O}(\lambda d^2)$

# Restricted Covariance Matrix

Restrict the covariance matrix to represent it with a linear number of parameter

- $\mathbf{C} = \mathbf{D}$, where $\mathbf{D}$ is diagonal
  $\Longrightarrow$ invariant to scaling of each coordinate $x \mapsto \mathbf{D}x$,
  not invariant to rotation
- $\mathbf{C} = \mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}}$, where $\mathbf{v} \in \mathbb{R}^d$
  $\Longrightarrow$ invariant to rotation of the search space $x \mapsto Rx$,
  not invariant to scaling

Our restriction:

$$\mathbf{C} = \mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D}$$

The parameter is $\theta = (\mathbf{m}, \sigma, \mathbf{v}, \mathbf{D}) \in \mathbb{R}^{3d+1}$.

# Required Modification

**$\mathcal{O}(d)$ sampling from $\mathcal{N}(\mathbf{m}, \sigma\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$ is required**

1. $\sqrt{\mathbf{C}^{(t)}} = \textsc{MatrixSqrt}(\mathbf{C}^{(t)})$ (perform every $\mathcal{O}(d/\lambda)$ iter.) $\mathcal{O}(\lambda d^2)$
2. $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 1, \ldots, \lambda$
3. $x_i = \mathbf{m}^{(t)} + \sigma^{(t)}\sqrt{\mathbf{C}^{(t)}} z_i$

4. $(x_{i:\lambda})_{i=1,\ldots,\lambda} = \textsc{SortW.r.t.}^f((x_i)_{i=1,\ldots,\lambda})$
5. $\mathbf{p}_{\mathbf{C}}^{(t+1)} = (1 - c_c)\mathbf{p}_{\mathbf{C}}^{(t)} + \sqrt{c_c(2 - c_c)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i (x_{i:\lambda} - \mathbf{m}^{(t)})/\sigma^{(t)}$
6. $\mathbf{p}_{\sigma}^{(t+1)} = (1 - c_\sigma)\mathbf{p}_{\sigma}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i z_{i:\lambda}$ $\mathcal{O}(\lambda d)$
7. $\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(t+1)}\|}{\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|]} - 1\right)\right)$
8. $\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{m}} l(\theta^{(t)}; x_{i:\lambda})$

**$\mathcal{O}(d)$ update for $\mathbf{v}$ and $\mathbf{D}$ instead of $\mathbf{C}$ is required**

9. $\mathbf{C}^{(t+1)} =$ $\mathbf{C}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{p}_{\mathbf{C}}^{(t+1)})$ $\mathcal{O}(\lambda d^2)$

# Sampling in $\mathcal{O}(d)$

**$\mathcal{O}(d)$ sampling from $\mathcal{N}(\mathbf{m}, \sigma\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$**

1. $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 1, \dots, \lambda$
2. $y_i = z_i + (\sqrt{1 + \|\mathbf{v}\|^2} - 1)\langle z, \mathbf{v}\rangle \mathbf{v}$
3. $x_i = \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{D}^{(t)} y_i$

$\mathcal{O}(\lambda d)$

4. $(x_{i:\lambda})_{i=1,\dots,\lambda} = \text{SortW.r.t.}^f((x_i)_{i=1,\dots,\lambda})$
5. $\mathbf{p}_{\mathbf{C}}^{(t+1)} = (1 - c_c)\mathbf{p}_{\mathbf{C}}^{(t)} + \sqrt{c_c(2 - c_c)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i (x_{i:\lambda} - \mathbf{m}^{(t)})/\sigma^{(t)}$
6. $\mathbf{p}_{\sigma}^{(t+1)} = (1 - c_\sigma)\mathbf{p}_{\sigma}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/(\sum w_i^2)} \sum_{i=1}^{\lambda} w_i z_{i:\lambda}$
7. $\sigma^{(t+1)} = \sigma^{(t)} \exp(\frac{c_\sigma}{d_\sigma}(\frac{\|\mathbf{p}_{\sigma}^{(t+1)}\|}{\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|]} - 1))$
8. $\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{m}} l(\theta^{(t)}; x_{i:\lambda})$

$\mathcal{O}(\lambda d)$

**$\mathcal{O}(d)$ update for $\mathbf{v}$ and $\mathbf{D}$ instead of $\mathbf{C}$ is required**

9. $\mathbf{C}^{(t+1)} =$
$\mathbf{C}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{p}_{\mathbf{C}}^{(t+1)})$

$\mathcal{O}(\lambda d^2)$

# Update of $\mathbf{v}$ and $\mathbf{D}$

CMA-ES:

$$\mathbf{C}^{(t+1)} = \mathbf{C}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

- $\tilde{\nabla}_{\mathbf{C}} l(\theta; x)$: natural gradient of the log-likelihood
- $\mathbf{p}_{\mathbf{C}}^{(t+1)}$: accumulation of successful steps in the past

Our approach:

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

$$\mathbf{D}^{(t+1)} = \mathbf{D}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

# Update of $\mathbf{v}$ and $\mathbf{D}$

CMA-ES:

$$\mathbf{C}^{(t+1)} = \mathbf{C}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{C}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

- $\tilde{\nabla}_{\mathbf{C}} l(\theta; x)$: natural gradient of the log-likelihood
- $\mathbf{p}_{\mathbf{C}}^{(t+1)}$: accumulation of successful steps in the past

Our approach:

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

$$\mathbf{D}^{(t+1)} = \mathbf{D}^{(t)} + c_\mu \sum_{i=1}^{\lambda} w_i \tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; x_{i:\lambda}) + c_1 \tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)} \mathbf{p}_{\mathbf{C}}^{(t+1)})$$

# Natural Gradient: Definition

The Natural Gradient $\tilde{\nabla} g(\theta)$ is the steepest direction of $g$ w.r.t. KL divergence

$$KL(p_\theta \parallel p_{\theta'}) := \int \ln \frac{p_\theta(x)}{p_{\theta'}(x)} p_\theta(x) \mathrm{d}x \approx \frac{1}{2}(\theta' - \theta)^{\mathrm{T}} G_\theta (\theta' - \theta) \ ,$$

where $G_\theta$ is the Fisher information matrix

$$G_\theta = \int \nabla l(\theta; x) \nabla l(\theta; x)^{\mathrm{T}} p_\theta(x) \mathrm{d}x, \quad \text{where } l(\theta; x) = \ln(p_\theta(x)) \ .$$

The natural gradient $\tilde{\nabla} g$ is given by

$$\tilde{\nabla} g(\theta) = G_\theta^{-1} \nabla g(\theta)$$

# Natural Gradient Computation $\tilde{\nabla} l(\theta; x) = G_\theta^{-1} \nabla l(\theta; x)$

- In a naive way, it requires cubic time in the number of *parameters*
  - E.g., for Gaussian with full covariance matrix $\mathbf{C}$, in which the number of parameters is $\mathcal{O}(d^2)$, then $G_\theta$ is of dimension $\mathcal{O}(d^2)$ and its inversion requires $\mathcal{O}(d^6)$
- For $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C})$ with $\theta = (\mathbf{m}, \mathbf{C})$, $\tilde{\nabla} l(\theta; x)$ has a closed form

$$\tilde{\nabla} l(\theta; x) = \begin{bmatrix} \tilde{\nabla}_{\mathbf{m}} l(\theta; x) \\ \tilde{\nabla}_{\mathbf{C}} l(\theta; x) \end{bmatrix} = \begin{bmatrix} x - \mathbf{m} \\ (x - \mathbf{m})(x - \mathbf{m})^{\mathrm{T}} / \sigma^2 - \mathbf{C} \end{bmatrix}$$

  and it requires only $\mathcal{O}(d^2)$.
- To compute the natural gradient for $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$ efficiently, we need to compute $\tilde{\nabla} l(\theta; x)$ directly (*without computing the inverse of Fisher*)

# Natural Gradient Computation $\tilde{\nabla} l(\theta; x) = G_\theta^{-1} \nabla l(\theta; x)$

- In a naive way, it requires cubic time in the number of *parameters*
  - E.g., for Gaussian with full covariance matrix $\mathbf{C}$, in which the number of parameters is $\mathcal{O}(d^2)$, then $G_\theta$ is of dimension $\mathcal{O}(d^2)$ and its inversion requires $\mathcal{O}(d^6)$
- For $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C})$ with $\theta = (\mathbf{m}, \mathbf{C})$, $\tilde{\nabla} l(\theta; x)$ has a closed form

$$\tilde{\nabla} l(\theta; x) = \begin{bmatrix} \tilde{\nabla}_{\mathbf{m}} l(\theta; x) \\ \tilde{\nabla}_{\mathbf{C}} l(\theta; x) \end{bmatrix} = \begin{bmatrix} x - \mathbf{m} \\ (x - \mathbf{m})(x - \mathbf{m})^{\mathrm{T}} / \sigma^2 - \mathbf{C} \end{bmatrix}$$

  and it requires only $\mathcal{O}(d^2)$.
- To compute the natural gradient for $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$ efficiently, we need to compute $\tilde{\nabla} l(\theta; x)$ directly (*without computing the inverse of Fisher*)

## Natural Gradient Computation $\tilde{\nabla} l(\theta; x) = G_\theta^{-1} \nabla l(\theta; x)$

- In a naive way, it requires cubic time in the number of *parameters*
  - E.g., for Gaussian with full covariance matrix $\mathbf{C}$, in which the number of parameters is $\mathcal{O}(d^2)$, then $G_\theta$ is of dimension $\mathcal{O}(d^2)$ and its inversion requires $\mathcal{O}(d^6)$
- For $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C})$ with $\theta = (\mathbf{m}, \mathbf{C})$, $\tilde{\nabla} l(\theta; x)$ has a closed form

$$\tilde{\nabla} l(\theta; x) = \begin{bmatrix} \tilde{\nabla}_{\mathbf{m}} l(\theta; x) \\ \tilde{\nabla}_{\mathbf{C}} l(\theta; x) \end{bmatrix} = \begin{bmatrix} x - \mathbf{m} \\ (x - \mathbf{m})(x - \mathbf{m})^{\mathrm{T}} / \sigma^2 - \mathbf{C} \end{bmatrix}$$

  and it requires only $\mathcal{O}(d^2)$.
- To compute the natural gradient for $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$ efficiently, we need to compute $\tilde{\nabla} l(\theta; x)$ directly (*without computing the inverse of Fisher*)

# $\mathcal{O}(d)$ Natural Gradient Computation

## $\mathcal{O}(d)$ Natural Gradient Computation

Let $\odot$ denote the element-wise multiplication. Let $\bar{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$, $\bar{V} = \operatorname{diag}(\bar{\mathbf{v}})$, $\bar{\bar{\mathbf{v}}} = \bar{\mathbf{v}} \odot \bar{\mathbf{v}}$, $\gamma_{\mathbf{v}} = 1 + \|\mathbf{v}\|^2$,
$\alpha = \min\Big(1, [\|\mathbf{v}\|^4 + (2 - \gamma_{\mathbf{v}}^{-1/2})\gamma_{\mathbf{v}} / \max_i(\bar{\bar{\mathbf{v}}}_i)]^{1/2}/(2 + \|\mathbf{v}\|^2)\Big)$,
$b = -(1-\alpha^2)\|\mathbf{v}\|^4\gamma_{\mathbf{v}}^{-1} + 2\alpha^2$, $A = 2\mathbf{I} - (b + 2\alpha)\bar{V}^2$. Compute

1. $s \leftarrow y \odot y - \|\mathbf{v}\|^2 \langle y, \bar{\mathbf{v}}\rangle \gamma_{\mathbf{v}}^{-1} y \odot \bar{\mathbf{v}} - \mathbf{1}$
2. $t \leftarrow \langle y, \bar{\mathbf{v}}\rangle y - 2^{-1}(\langle y, \bar{\mathbf{v}}\rangle^2 + \gamma_{\mathbf{v}})\bar{\mathbf{v}}$
3. $s \leftarrow s - \alpha\gamma_{\mathbf{v}}^{-1}\left((2 + \|\mathbf{v}\|^2)\bar{\mathbf{v}} \odot t - \|\mathbf{v}\|^2 \langle \bar{\mathbf{v}}, t\rangle \bar{\bar{\mathbf{v}}}\right)$
4. $s \leftarrow A^{-1}s - \left(1 + b\langle \bar{\bar{\mathbf{v}}}, A^{-1}\bar{\bar{\mathbf{v}}}\rangle\right)^{-1} b\langle s, A^{-1}\bar{\bar{\mathbf{v}}}\rangle A^{-1}\bar{\bar{\mathbf{v}}}$
5. $t \leftarrow t - \alpha\left[(2 + \|\mathbf{v}\|^2)\bar{\mathbf{v}} \odot s - \langle s, \bar{\bar{\mathbf{v}}}\rangle \bar{\mathbf{v}}\right]$

Then, $\tilde{\nabla}_{\mathbf{v}} l(\theta; x) = \|\mathbf{v}\|^{-1} t$ and $\tilde{\nabla}_{\mathbf{D}} l(\theta; x) = \mathbf{D}s$.

# VD-CMA: A Linear Time/Space Variant of CMA

## $\mathcal{O}(d)$ sampling from $\mathcal{N}(\mathbf{m}, \sigma\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D})$

1. $z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 1, \dots, \lambda$
2. $y_i = z_i + ((1 + \|\mathbf{v}\|^2)^{1/2} - 1)\langle z, \mathbf{v}\rangle\mathbf{v}$
3. $x_i = \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{D}^{(t)}y_i$

4. $(x_{i:\lambda})_{i=1,\dots,\lambda} = \textsc{SortW.r.t.}^f((x_i)_{i=1,\dots,\lambda})$
5. $\mathbf{p}_{\mathbf{C}}^{(t+1)} = (1 - c_c)\mathbf{p}_{\mathbf{C}}^{(t)} + \sqrt{c_c(2 - c_c)/(\sum w_i^2)}\sum_{i=1}^{\lambda} w_i(x_{i:\lambda} - \mathbf{m}^{(t)})/\sigma^{(t)}$
6. $\mathbf{p}_{\sigma}^{(t+1)} = (1 - c_\sigma)\mathbf{p}_{\sigma}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/(\sum w_i^2)}\sum_{i=1}^{\lambda} w_i z_{i:\lambda}$
7. $\sigma^{(t+1)} = \sigma^{(t)}\exp(\frac{c_\sigma}{d_\sigma}(\frac{\|\mathbf{p}_\sigma^{(t+1)}\|}{\mathbb{E}[\|\mathcal{N}(\mathbf{0},\mathbf{I})\|]} - 1))$
8. $\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \sum_{i=1}^{\lambda} w_i\tilde{\nabla}_{\mathbf{m}} l(\theta^{(t)}; x_{i:\lambda})$

## $\mathcal{O}(d)$ update for $\mathbf{v}$ and $\mathbf{D}$

9. $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + c_\mu\sum_{i=1}^{\lambda} w_i\tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; x_{i:\lambda}) + c_1\tilde{\nabla}_{\mathbf{v}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{p}_{\mathbf{C}}^{(t+1)})$
10. $\mathbf{D}^{(t+1)} = \mathbf{D}^{(t)} + c_\mu\sum_{i=1}^{\lambda} w_i\tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; x_{i:\lambda}) + c_1\tilde{\nabla}_{\mathbf{D}} l(\theta^{(t)}; \mathbf{m}^{(t)} + \sigma^{(t)}\mathbf{p}_{\mathbf{C}}^{(t+1)})$

# Testbed

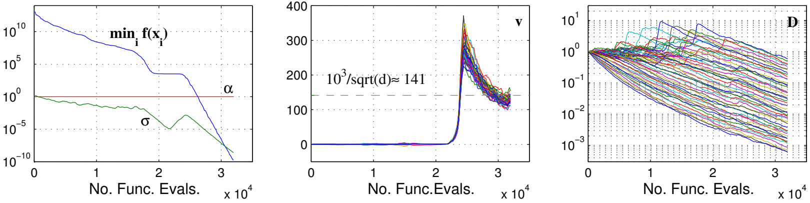
| | |
|---|---|
| Sphere | $f_{\text{sph}}(x) = x^{\mathrm{T}}\mathbf{I}x$ |
| Tablet | $f_{\text{tab}}(x) = x^{\mathrm{T}}D_{\text{tab}}x, \qquad D_{\text{tab}} = \mathrm{diag}(10^6, 1, \dots, 1)$ |
| Ellipsoid | $f_{\text{ell}}(x) = x^{\mathrm{T}}D_{\text{ell}}x, \qquad D_{\text{ell}} = \mathrm{diag}(10^{3\frac{0}{d-1}}, \dots, 10^{3\frac{d-1}{d-1}})$ |
| Cigar | $f_{\text{cig}}(x) = x^{\mathrm{T}}D_{\text{cig}}x, \qquad D_{\text{cig}} = \mathrm{diag}(1, 10^6, \dots, 10^6)$ |
| Rot-Cigar | $f_{\text{cigrot}}(x) = x^{\mathrm{T}}(10^6\mathbf{I} + (1 - 10^6)uu^{\mathrm{T}})x$ |
| Ellipsoid-Cigar | $f_{\text{ellcig}}(x) = x^{\mathrm{T}}D_{\text{ell}}(10^6\mathbf{I} + (1 - 10^6)uu^{\mathrm{T}})D_{\text{ell}}x$ |
| Rot-Tablet | $f_{\text{tabrot}}(x) = x^{\mathrm{T}}(\mathbf{I} + (10^6 - 1)uu^{\mathrm{T}})x$ |
| Rot-Ellipsoid | $f_{\text{ellrot}}(x) = x^{\mathrm{T}}R^{\mathrm{T}}D_{\text{ell}}Rx$ |
| Rosenbrock | $f_{\text{ros}}(x) = \sum_{i=1}^{d-1} 10^2(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$ |
| Rot-Rosenbrock | $f_{\text{rosrot}}(x) = f_{\text{ros}}(Rx)$ |

- $R$ is an orthogonal matrix and $u$ is a unit vector, both are randomly generated for each run.
- Target function value is $10^{-10}$; Maximum number of FEs is $5 \cdot 10^7$.
- The inverse Hessian of $f_{\text{tabrot}}$ and $f_{\text{ellrot}}$ are not of the form $\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D}$

# On 50 dimensional $f_{\text{ellcig}}$

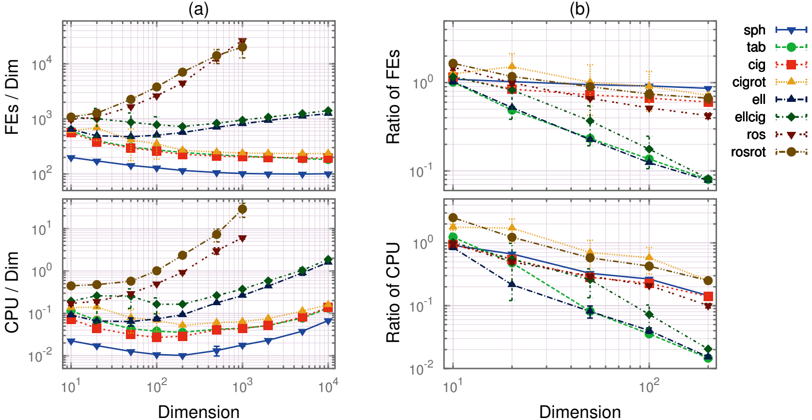
$$f_{\text{ellcig}}(x) = x^{\mathrm{T}} D_{\text{ell}} (10^6 \mathbf{I} + (1 - 10^6) u u^{\mathrm{T}}) D_{\text{ell}} x$$

- $D_{\text{ell}} = \operatorname{diag}(1, \dots, 10^{3\frac{i-1}{d-1}}, \dots, 10^3)$
- $u = (1/\sqrt{d}, \dots, 1/\sqrt{d})$
- $Hess(f_{\text{ellcig}})^{-1} \propto D_{\text{ell}}^{-1} (\mathbf{I} + 10^6 u u^{\mathrm{T}}) D_{\text{ell}}^{-1}$




$\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}})\mathbf{D}$ becomes proportional to $Hess(f_{\text{ellcig}})^{-1}$ ($\mathbf{D} \propto D_{\text{ell}}^{-1}$, $\mathbf{v} \approx 10^3 u$), and speed up the convergence afterwords.

# VS. CMA (averaged over 10 independent runs)

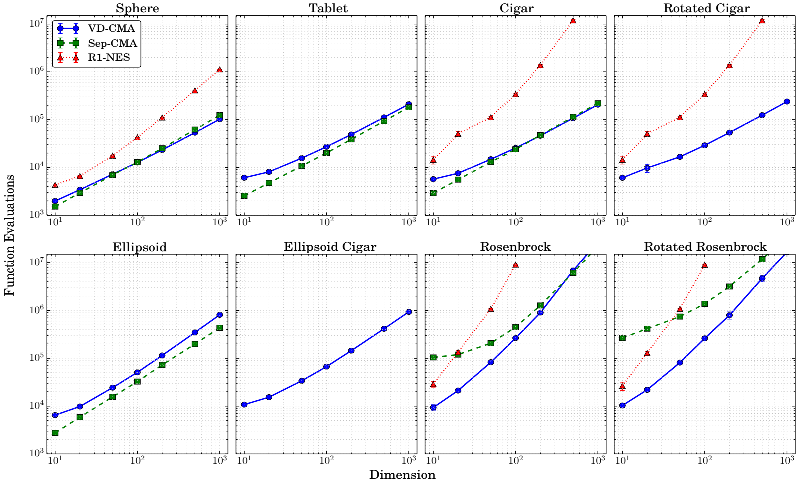



(a) Number of function evaluations (FEs) and CPU time [s] divided by $d$.

(b) FEs and CPU time spent by VD-CMA divided by those spent by the CMA.

- Needs more than $5d \times 10^4$ FEs on $f_{\text{tabrot}}$ and $f_{\text{ellrot}}$.
- Faster in CPU and even in FEs on other functions due to larger $c_\mu$ and $c_1$.

# VS Other Linear Time Variants




Sep-CMA: covariance matrix restricted to diagonal, $\mathbf{C} = \mathbf{D}$

R1-NES: covariance matrix restricted to $\mathbf{C} = \mathbf{I} + \mathbf{v}\mathbf{v}^{\mathrm{T}}$

Source code available at

- my homepage (Matlab/Octave)
  `https://sites.google.com/site/youheiakimotospage/`
- Shark Library in C++
  `http://image.diku.dk/shark`
- libcmaes in C++
  `https://github.com/beniz/libcmaes`

For more details, see

- Youhei Akimoto, Anne Auger, and Nikolaus Hansen. "Comparison-based Natural Gradient Optimization in High Dimension". In: *Proceedings of Genetic and Evolutionary Computation Conference*. 2014, pp. 373–380

# VD-CMA: Modified Fisher Information

To update $\mathbf{m}$, $\mathbf{v}$, $\mathbf{D}$ in $\mathcal{O}(d)$, we need to compute $\tilde{\nabla} l(\theta; x) = G_{\theta}^{-1} \nabla l(\theta; x)$ analytically, where

$$G_{\theta} = \begin{bmatrix} G_{\mathbf{m}} & 0 & 0 \\ 0 & G_{\mathbf{v},\mathbf{v}} & G_{\mathbf{v},\mathbf{D}} \\ 0 & G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{bmatrix}$$

Diagonal blocks are all nonsingular, however $\begin{bmatrix} G_{\mathbf{v},\mathbf{v}} & G_{\mathbf{v},\mathbf{D}} \\ G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{bmatrix}$ is sometimes singular. Then the natural gradient is not well-defined and the update becomes unstable.

We define the modified Fisher information and define the natural gradient using this:

$$G_{\theta} = \begin{bmatrix} G_{\mathbf{m}} & 0 & 0 \\ 0 & G_{\mathbf{v},\mathbf{v}} & \alpha G_{\mathbf{v},\mathbf{D}} \\ 0 & \alpha G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{bmatrix}$$

Then, for an appropriate $\alpha \in [0, 1]$, $G_{\theta}$ is guaranteed to be nonsingular.

# VD-CMA: Modified Fisher Information

To update $\mathbf{m}$, $\mathbf{v}$, $\mathbf{D}$ in $\mathcal{O}(d)$, we need to compute $\tilde{\nabla} l(\theta; x) = G_\theta^{-1} \nabla l(\theta; x)$ analytically, where

$$G_\theta = \begin{bmatrix} G_{\mathbf{m}} & 0 & 0 \\ 0 & G_{\mathbf{v},\mathbf{v}} & G_{\mathbf{v},\mathbf{D}} \\ 0 & G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{bmatrix}$$

Diagonal blocks are all nonsingular, however $\left[\begin{smallmatrix} G_{\mathbf{v},\mathbf{v}} & G_{\mathbf{v},\mathbf{D}} \\ G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{smallmatrix}\right]$ is sometimes singular. Then the natural gradient is not well-defined and the update becomes unstable.

We define the modified Fisher information and define the natural gradient using this:

$$G_\theta = \begin{bmatrix} G_{\mathbf{m}} & 0 & 0 \\ 0 & G_{\mathbf{v},\mathbf{v}} & \alpha G_{\mathbf{v},\mathbf{D}} \\ 0 & \alpha G_{\mathbf{v},\mathbf{D}}^{\mathrm{T}} & G_{\mathbf{D},\mathbf{D}} \end{bmatrix}$$

Then, for an appropriate $\alpha \in [0, 1]$, $G_\theta$ is guaranteed to be nonsingular.