Adaptive Coordinate Descent

Ilya Loshchilov[1,2], Marc Schoenauer[1,2], Michèle Sebag[2,1]

[1]TAO Project-team, INRIA Saclay - Île-de-France

[2]and Laboratoire de Recherche en Informatique (UMR CNRS 8623)
Université Paris-Sud, 91128 Orsay Cedex, France

Content

Motivation

Coordinate Descent

- Fast and Simple
- Not suitable for non-separable optimization

A hypothesis to check

- Some Coordinate Descent with adaptation of coordinate system can be as fast as the state-of-the art Evolutionary Algorithms.

Covariance Matrix Adaptation Evolution Strategy

Decompose to understand

- While CMA-ES by definition is CMA and ES, only recently the **algorithmic decomposition** has been presented. [1]

Algorithm 1 CMA-ES = Adaptive Encoding + ES

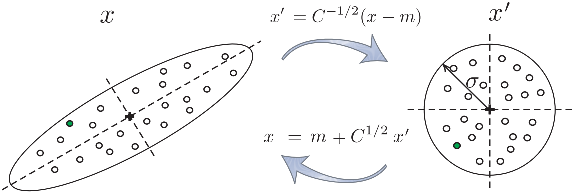
1: $x_i \leftarrow m + \sigma \mathcal{N}_i(0, \boldsymbol{I})$, for $i = 1 \ldots \lambda$
2: $f_i \leftarrow f(\mathbf{B}x_i)$, for $i = 1 \ldots \lambda$
3: **if** Evolution Strategy (ES) with 1/5th success rule **then**
4: $\quad \sigma \leftarrow \sigma exp^{\propto(\frac{\texttt{success rate}}{\texttt{expected success rate}} - 1)}$
5: **if** Cumulative Step-Size Adaptation ES (CSA-ES) **then**
6: $\quad \sigma \leftarrow \sigma exp^{\propto(\frac{\|\texttt{evolution path}\|}{\|\texttt{expected evolution path}\|} - 1)}$
7: $\mathbf{B} \leftarrow$ AdaptiveEncoding$(\mathbf{B}x_1, \ldots, \mathbf{B}x_\mu)$

[1] N. Hansen (2008). "Adaptive Encoding: How to Render Search Coordinate System Invariant"

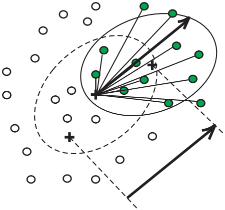
Adaptive Encoding

Inspired by Principal Component Analysis (PCA)

Principal Component Analysis



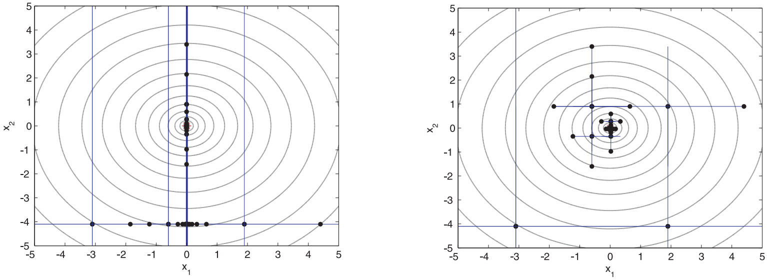
Adaptive Encoding Update



Coordinate Descent (CD)

Simple Idea

- **Goal**: minimize $f(x) = {x_1}^2 + {x_2}^2$ up to value $f_{target} = 10^{-10}$
- **Initial state**: $x_0 = (-3.1, -4.1)$, step-size $\sigma = 10$
- **Simple Idea**: iteratively optimize $f(x)$ with respect to one coordinate, while other are fixed



(a) Optimize the first coordinate by Dichotomy, then the second.

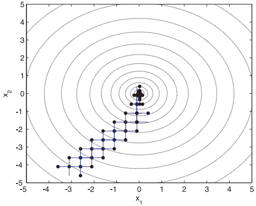
(b) Cyclically optimize the first and the second coordinates.

Coordinate Descent (CD)

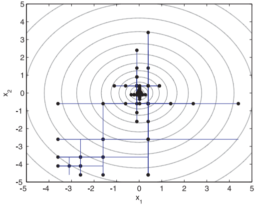
Algorithm

Algorithm 1 Coordinate Descent

$m \leftarrow x^{min}_{i:d} + \mathbb{U}_{i:d}(x^{max}_{i:d} - x^{min}_{i:d})$
$f_{best} \leftarrow evaluate(m)$
$\sigma_{i:d} \leftarrow (x^{max}_{i:d} - x^{min}_{i:d})/4$
$i_x \leftarrow 0$
while NOT Stopping Criterion **do**
  $i_x \leftarrow i_x + 1$ mod. d // Cycling over $[1, d]$
  $x'_{1:d} \leftarrow 0$
  $x'_{i_x} \leftarrow -\sigma_{i_x}$; $x_1 \leftarrow m + x'$; $f_1 \leftarrow evaluate(x_1)$
  $x'_{i_x} \leftarrow +\sigma_{i_x}$; $x_2 \leftarrow m + x'$; $f_2 \leftarrow evaluate(x_2)$
  $succ \leftarrow 0$
  if $f_1 < f_{best}$ **then**
    $f_{best} \leftarrow f_1$; $m \leftarrow x_1$; $succ \leftarrow 1$
  if $f_2 < f_{best}$ **then**
    $f_{best} \leftarrow f_2$; $m \leftarrow x_2$; $succ \leftarrow 1$
  if $succ = 1$ **then**
    $\sigma_{i_x} \leftarrow k_{succ} \cdot \sigma_{i_x}$
  else
    $\sigma_{i_x} \leftarrow k_{unsucc} \cdot \sigma_{i_x}$



(b) $k_{succ} = 1.0$, 117 evals.



(b) $k_{succ} = 2.0$, 149 evals.

Coordinate Descent (CD)

Convergence Rates

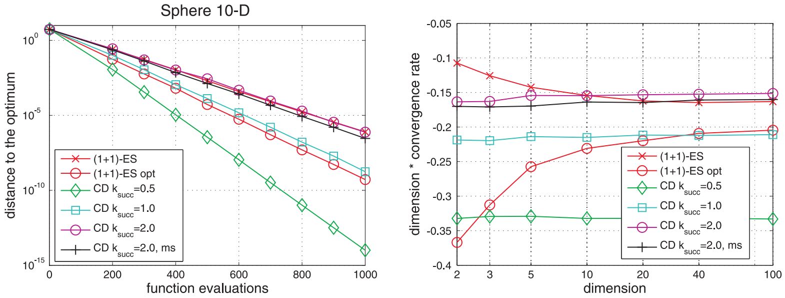


Figure: **Left**: Evolution of distance to the optimum versus number of function evaluations for the (1+1)-ES, (1+1)-ES opt, CD $k_{succ} = 0.5$, CD $k_{succ} = 1.0$, CD $k_{succ} = 2.0$ and CD $k_{succ} = 2.0$ ms on $f(x) = \|x\|^2$ in dimension **10**. **Right**: Convergence rate c (the lower the better) multiplied by the dimension d for different algorithms depending on the dimension d. The convergence rates have been estimated for the median of 101 runs.

Adaptive Coordinate Descent (ACiD)

Coordinate Descent for optimization of non-separable problems

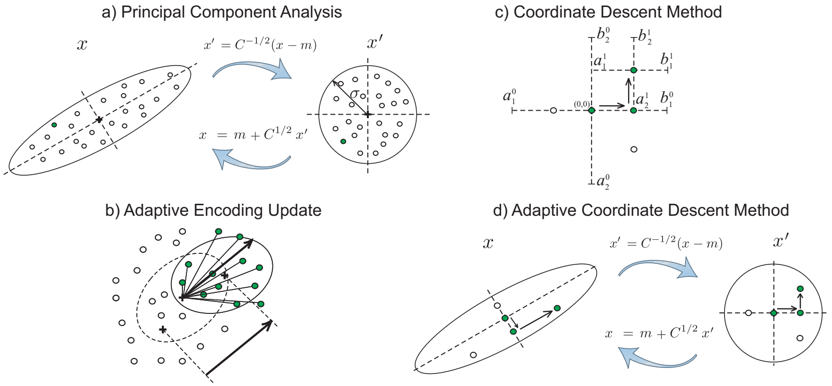


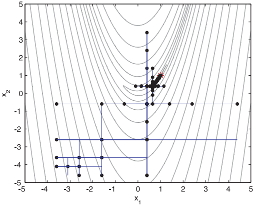
Figure: AE$_{CMA}$-like Adaptive Encoding Update (**b**) mostly based on Principal Component Analysis (**a**) is used to extend some Coordinate Descent method (**c**) to the optimization of non-separable problems (**d**).

Adaptive Coordinate Descent (ACiD)

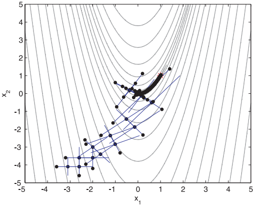
ACiD on Rosenbrock 2-D function

Algorithm 1 Adaptive Coordinate Descent (ACiD)

$m \leftarrow x^{min}_{i:d} + \mathbb{U}_{i:d}(x^{max}_{i:d} - x^{min}_{i:d})$
$f_{best} \leftarrow evaluate(m)$
$\sigma_{i:d} \leftarrow (x^{max}_{i:d} - x^{min}_{i:d})/4$
$\mathbf{B} \leftarrow \boldsymbol{I}$
$i_x \leftarrow 0$
while NOT Stopping Criterion **do**
  $i_x \leftarrow i_x + 1$ mod. d // Cycling over $[1, d]$
  $x'_{1:d} \leftarrow 0$
  $x'_{i_x} \leftarrow -\sigma_{i_x}$; $x_1 \leftarrow m + \mathbf{B}x'$; $f_1 \leftarrow evaluate(x_1)$
  $x'_{i_x} \leftarrow +\sigma_{i_x}$; $x_2 \leftarrow m + \mathbf{B}x'$; $f_2 \leftarrow evaluate(x_2)$
  $succ \leftarrow 0$
  if $f_1 < f_{best}$ **then**
    $f_{best} \leftarrow f_1$; $m \leftarrow x_1$; $succ \leftarrow 1$
  if $f_2 < f_{best}$ **then**
    $f_{best} \leftarrow f_2$; $m \leftarrow x_2$; $succ \leftarrow 1$
  if $succ = 1$ **then**
    $\sigma_{i_x} \leftarrow k_{succ} \cdot \sigma_{i_x}$
  else
    $\sigma_{i_x} \leftarrow k_{unsucc} \cdot \sigma_{i_x}$
  $x^a_{(2i_x-1)} \leftarrow x_1$; $f^a_{(2i_x-1)} \leftarrow f_1$
  $x^a_{2i_x} \leftarrow x_2$; $f^a_{2i_x} \leftarrow f_2$
  if $i_x = d$ **then**
    $x^a \leftarrow \left\{ x^a_{< f^a_i : i} \middle| 1 \le i \le 2d \right\}$
    $\mathbf{B} \leftarrow AdaptiveEncoding(x^a_1, \ldots, x^a_\mu)$



(b) $k_{succ} = 2.0$, 22231 evals.



(b) $k_{succ} = 1.2$, 325 evals.

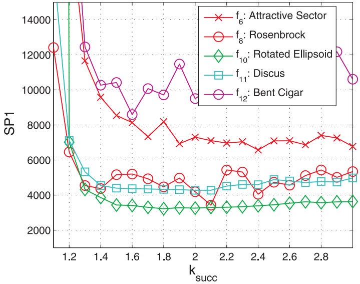
Experimental Validation

ACiD on the noiseless Black-Box Optimization Benchmarking (BBOB) testbed

BBOB benchmarking

- 24 uni-modal, multi-modal, ill-conditioned, separable and non-separable problems.
- Results available of BIPOP-CMA-ES, IPOP-CMA-ES, (1+1)-CMA-ES and many other state-of-the-art algorithms.

- How ACiD is sensitive to the step-size multiplier k_{succ} ? (an example in 10-D)



Experimental Validation

Comparison of ACiD and (1+1)-CMA-ES

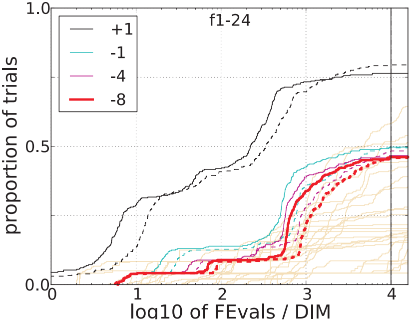


Figure: BBOB-like results for noiseless functions in **20-D**: Empirical Cumulative Distribution Function (ECDF), for **ACiD (continuous lines)** and $(1+1)$**-CMA-ES (dashed lines)**, of the running time (number of function evaluations), normalized by dimension d, needed to reach target precision $f_{opt} + 10^k$ (for $k = +1, -1, -4, -8$). Light yellow lines in the background show similar ECDFs for target value 10^{-8} of all algorithms benchmarked during BBOB 2009.

Experimental Validation

Comparison with IPOP-CMA-ES, BIPOP-CMA-ES and (1+1)-CMA-ES

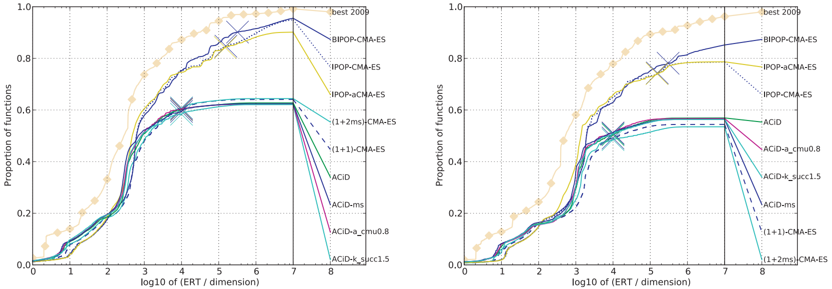
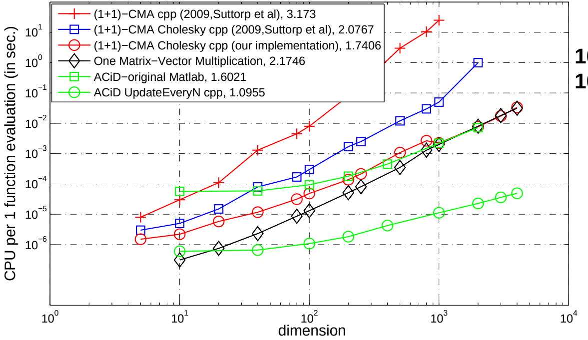


Figure: Empirical cumulative distribution of the bootstrapped distribution of ERT vs dimension for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in **10-D** (**Left**) and **40-D** (**Right**). The best ever line corresponds to the best result obtained by at least one algorithm from BBOB 2009 for each of the targets.

Computation Complexity

Some work in progress

- Eigendecomposition as a part of CMA-like algorithms: $O(n^3)$.
- Cholesky factor and its inverse can be learned incrementally: $O(n^2)$.
- (1+1)-CMA-ES has $O(n^3)$; with Cholesky update - $O(n^2)$.
- ACiD has $O(n^2)$; might have $O(n)$.



100.000 evaluations for **1000**-dimensional **Ellipsoid**:

- CMA-ES - 15 minutes. (GECCO 2011 CMA-ES Tutorial)
- ACiD - 1-2 seconds. ($f = 10^{-8}$ after 5-10 minutes and $2.4 \cdot 10^7$ evaluations)

Summary

ACiD

- At least as fast as (1+1)-CMA-ES.
- The computation complexity is $O(n^2)$, might be $O(n)$ if call eigendecomposition every n iterations.
- The source code is available online: `http://www.lri.fr/~ilya/publications/ACiDgecco2011.zip`

Open Questions

- $O(n)$ complexity, that is important for large-scale optimization in Machine Learning.
- Extention to multi-modal optimization.
- Fast meta-models in dimension $d < n$ (even for d=1).

Summary

Thank you for your attention!

Questions?

Adaptive Encoding

Algorithm 1 Adaptive Encoding

Input: $x_1, \ldots, x_\mu$
if *Initialize* **then**
  $w_i \leftarrow \frac{1}{\mu}$; $c_p \leftarrow \frac{1}{\sqrt{d}}$; $c_1 \leftarrow \frac{0.5}{d}$; $c_\mu \leftarrow \frac{0.5}{d}$
  $\mathbf{p} \leftarrow 0$
  $\mathbf{C} \leftarrow \boldsymbol{I}$; $\mathbf{B} \leftarrow \boldsymbol{I}$
  $m \leftarrow \sum_{i=1}^{\mu} x_i w_i$
  return.
$m^- \leftarrow m$
$m \leftarrow \sum_{i=1}^{\mu} x_i w_i$
$z_0 \leftarrow \frac{\sqrt{d}}{\|\mathbf{B}^{-1}(m - m^-)\|}(m - m^-)$
$z_i \leftarrow \frac{\sqrt{d}}{\|\mathbf{B}^{-1}(x_i - m^-)\|}(x_i - m^-)$
$\mathbf{p} \leftarrow (1 - c_p)\mathbf{p} + \sqrt{c_p(2 - c_p)} z_0$
$\mathbf{C}_\mu \leftarrow \sum_{i=1}^{\mu} w_i z_i z_i^T$
$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\mathbf{C} + c_1 \mathbf{p}\mathbf{p}^T + c_\mu \mathbf{C}_\mu$
$\mathbf{B}^\circ \mathbf{D}\mathbf{D}\mathbf{B}^\circ \leftarrow \texttt{eigendecomposition}(\mathbf{C})$
$\mathbf{B} \leftarrow \mathbf{B}^\circ \mathbf{D}$
Output: $\mathbf{B}$