(μ, λ)-CCMA-ES

for constrained black-box optimization

Daniel Wälchli

Computational Science and Engineering Lab
ETH Zürich
With:

Georgios Arampatzis, Pascal Weber, Henri Rästas, Petros Koumoutsakos (all CSELab)
Motivation

Objective:

\[ x^* = \min_{x \in \Omega} f(x) \]

for \( \Omega \subset \mathbb{R}^n \), under the inequality constraints

\[ h_j(x) \leq 0, \quad j \in \{1, \ldots, m\} \]

Black-Box Optimization: no assumptions on the analytic form of the objective function \( f(x) \), no access to gradients of \( f(x) \)

Constraints: here also part of the black-box setting, i.e. no access to derivatives of constraints \( h_j(x) \)
Motivation (cont.)

(constrained) optimization is ubiquitous, e.g:

- Mixing of Fluids
  - [Link](https://www.colorado.edu/center/aerospace/mechanics/research/multi-physics-design-optimization)

- Cell Separation
  - [Link](https://www.cse-lab.ethz.ch/research/projects/life-sciences)

- Drag Reduction
  - [Link](https://www.pinterest.ch/milanrohrer/engineering/)
Background


• **Viability Boundaries**: Viability Principles for Constrained Optimization Using a (1+1)-CMA-ES by Andreas Maesani and Dario Floreano [2014]. *Parallel Problem Solving from Nature – PPSN XIII*. Springer International Publishing, Cham, 272–281

(μ, λ)-CMA-ES

Algorithm 1 CMA-ES

1. Initialize algorithm ▶ fig. 1a
2. while Termination criteria not met do
3. Sampling \( x_i \sim \mathcal{N}(\mu^{(g)}, \Sigma^{(g)}) \) ▶ fig. 1b
4. Evaluate individual fitness \( f(x_i) \)
5. Selection and recombination ▶ fig. 1c
6. Adaptation \( \mu^{(g+1)} \) and \( \Sigma^{(g+1)} \) ▶ fig. 1d
7. end while
8. Return best ever found \( x^* \) and \( f(x^*) \)


Example: \((\mu, \lambda)\)-CMA-ES
Algorithm 2 CCMA-ES

1: Initialize algorithm
2: while Termination criteria not met do
3:   If $\mu^{(g)}$ violates constraint $h_j$, update viability bounds $b_j$
4:   Sampling $x_i \sim \mathcal{N}(\mu^{(g)}, \Sigma^{(g)})$
5:   If $x_i$ violates $h_j$, handle constraint and adapt $\Sigma^{(g)}$
6:   Evaluate individual fitness $f(x_i)$
7:   Selection and recombination
8:   Adaptation $\mu^{(g+1)}$ and $\Sigma^{(g+1)}$
9:   end while
10: Return best ever found $x^*$ and $f(x^*)$
Constraint Handling

Algorithm 3 Constraint Handling in CCMA-ES

1: while Constraints violated do
2:    for $i = 1, \ldots, \lambda$ do ▷ for all offspring individuals
3:        for $j = 1, \ldots, m$ do ▷ for all constraints
4:            if $h_j(x_i) > 0$ then ▷ if $x_i$ violates constraint
5:                $v_j \leftarrow (1 - c_v) v_j + c_v y_i$
6:                \[ C \leftarrow C - \frac{\beta}{\alpha_0(x_i) \| v_j \|^2} v_j v_j^T \]
7:            end if
8:        end for
9:    end for
10:   for $i = 1, \ldots, \lambda$ do ▷ for all offspring individuals
11:       for $j = 1, \ldots, m$ do ▷ for all constraints
12:           if $h_j(x_i) > 0$ then ▷ if $x_i$ violates constraint
13:               Resample offspring $x_i$
14:           end if
15:       end for
16:   end for
17: end while

Constraint Normal Approximation

\[ v_j = (1 - c_v) v_j + c_v y_i \]

Covariance Matrix Correction

\[ C = C - \frac{\beta}{\alpha_0(x)} \sum_{j=1}^{m} \alpha_j(x) \frac{v_j v_j^T}{\|v_j\|^2} \]
Constraint Handling (cont.)

Constraint Normal Approximation

\[ \nu_j = (1 - c_v)\nu_j + c_v y_i \]

Covariance Matrix Correction

\[ C = C - \frac{\beta}{\alpha_0(x)} \sum_{j=1}^{m} \alpha_j(x) \frac{\nu_j \nu_j^T}{\|\nu_j\|^2} \]
Viability Boundaries

**Problem**: it may be difficult to find starting point $m^{(g)}$ inside valid region (satisfying $h_j(m^{(g)}) \leq 0$)

**Solution**: introduce **viability boundaries**:

$$b = \left[ \max \{0, h_1(x_1), \ldots, h_1(x_\lambda)\}, \ldots, \max \{0, h_m(x_1), \ldots, h_m(x_\lambda)\} \right]$$

$b$ is a relaxed boundary, initialised to the largest constraint violation at start and $b$ is contracted at each generation until $b = 0$:

$$\min \left\{ b_i, h_i(x_{c,i}) + \frac{b_i - h_i(x_{c,i})}{2} \right\}, \text{where } x_{c,i} \text{ denotes the sample closest to } h_i$$
Discussion Viability Boundaries

fig. a: CCMA-ES initialized with mean $m^{(0)}$ outside valid domain
fig. b: Two samples created, both violating constraint $h$
fig. c: Relaxed boundary adapted to greatest constraint violation
fig. d: New proposal distribution calculated according to CMA-ES
Example: $(\mu, \lambda)$-CCMA-ES
Evaluation

2006 CEC Test Problems

speed-up measured in terms of function evaluations

Baseline: mVIE

mVIE outperforms various optimization algorithms, such as variants of Differential Evolution, CMA-ES, Particle Swarm Optimization and other (see reference)

Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

\[ x = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q}) \]

to reduce tumour size at \( t_{\text{end}} : P^* = P + Q + Q_D \)

With respect to constraints:

\[ h_j = a_j - 1 \leq 0, \quad j \in \{1, \ldots, n_q\} \]

\[ h_{\text{max}} = \max_t C(t) - v_{\text{max}} \leq 0 \]

\[ h_c = \int_0^{t_{\text{end}}} C(t)dt - v_{\text{cum}} \leq 0 \]
Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

\[ x = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q}) \]

to reduce tumour size at \( t_{\text{end}} : P^* = P + Q + Q_D \)

With respect to constraints:

\[ h_j = a_j - 1 \leq 0, \quad j \in \{1, \ldots, n_q\} \]

\[ h_{\text{max}} = \max_t C(t) - u_{\text{max}} \leq 0. \]

\[ h_c = \int_0^{t_{\text{end}}} C(t) dt - u_{\text{cum}} \leq 0 \]

\[
\begin{align*}
\frac{dC}{dt} &= -\theta_1 C \\
\frac{dP}{dt} &= \theta_4 P \left(1 - \frac{P + Q + Q_D}{K}\right) + \theta_5 Q_D - \theta_3 P - \theta_1 \theta_2 CP \\
\frac{dQ}{dt} &= \theta_3 P - \theta_1 \theta_2 CQ \\
\frac{dQ_D}{dt} &= \theta_1 \theta_2 CQ - \theta_3 Q_D - \theta_5 Q_D.
\end{align*}
\]
Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

\[ x = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q}) \]

to reduce tumour size at \( t_{\text{end}} \): \( P^* = P + Q + Q_D \)

With respect to constraints:

\[ h_j = a_j - 1 \leq 0, \quad j \in \{1, \ldots, n_q\} \]

\[ h_{\text{max}} = \max_t C(t) - \nu_{\text{max}} \leq 0. \]

\[ h_c = \int_{0}^{t_{\text{end}}} C(t)dt - \nu_{\text{cum}} \leq 0 \]
**Conclusion**

1. CCMA-ES is a novel optimization algorithm for constrained black box settings
2. CCMA-ES is outperforming state of the art methods
3. due to its algorithmic structure it is easily parallelizable and hence well suited for HPC applications
We at the CSE-Lab are building Korali, a high-performance computing framework for optimization and Bayesian uncertainty quantification of large-scale computational models.

Soon available here: https://github.com/cselab

Optimization:
• CMA-ES
• CCMA-ES
• Plotting & Analysis Tools
UQ:
• TMCMC

Support for parallel execution (pthreads, MPI, UPC++) and GPU based computational models
Backend implemented in C++
Interface for Python
Appendix
Algorithm 1 CMA-ES

1: Initialize algorithm
2: while Termination criteria not met do
3:   Sampling $x_i \sim N(m^{(g)}, \Sigma^{(g)})$
4:   Evaluate individual fitness $f(x_i)$
5:   Selection and recombination
6:   Adaptation $m^{(g+1)}$ and $\Sigma^{(g+1)}$
7: end while
8: Return best ever found $x^*$ and $f(x^*)$

**Evaluation.** The objective function $f$ is evaluated at the obtained individuals $x_i$ and sorted $f(x_{1:\lambda}) \leq f(x_{2:\lambda}) \leq \cdots \leq f(x_{\lambda:\lambda})$.

$$m^{g+1} = \sum_{i=1}^\mu w_i x_i^{g+1}_{\ell:\lambda}.$$  

$$C^{g+1} = (1-c_1-c_\mu)C^g + c_1 p_c^{g+1} (p_c^{g+1})^T + c_\mu \sum_{i=1}^\mu w_i y^{g+1}_{\ell:\lambda} (y^{g+1}_{\ell:\lambda})^T$$

with $p_c^{g+1} = (1-c_c)p_c^g + \sqrt{c_c(2-c_c)\mu_{\text{eff}}} \frac{m^{g+1} - m^g}{\sigma^g}$. 

$\mu$, $\lambda$-CMA-ES Updating Rule
g09: speed-up ~5.5 (best)

Minimize [3]:

\[ f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_6^2 + 7x_8^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \]

subject to:

\[ g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \]
\[ g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + 4x_4 - 5x_5 \leq 0 \]
\[ g_3(\vec{x}) = 196 + 23x_1 + x_2^2 + 6x_6 - 8x_7 \leq 0 \]
\[ g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \]

where \(-10 \leq x_i \leq 10\) for \(i = 1, \ldots, 7\). The optimum solution is \(\vec{x}^* = (2.33049935147405174, 1.95137238487114592, -0.477541399510615805, 1.36572624923625874, -0.624486595100388983, \).

g19: speed-up ~0.8 (worst)

Minimize [8]:

\[ f(\vec{x}) = \sum_{i=1}^{5} \sum_{j=1}^{5} c_{ij}x_{(10+i)}x_{(10+j)} + 2 \sum_{j=1}^{5} d_{j}x_{(10+j)} - \sum_{i=1}^{5} b_{i}x_{i} \]  

subject to:

\[ g_1(\vec{x}) = -2 \sum_{i=1}^{5} c_{ij}x_{(10+i)} - 3d_{j}x_{(10+j)} + c_{j} + \sum_{i=1}^{5} a_{ij}x_{i} \leq 0 \quad j = 1, \ldots, 5 \]

where \(\delta = [-40, -2, -0.25, -4, -1, -40, -60, 5, 1]\) and the remaining data is on Table 1. The bounds are \(0 \leq x_i \leq 10\) for \(i = 1, \ldots, 15\). The best known solution is at \(\vec{x}^* = (1.6069134129291344e-17, 3.9537829854150509e-16, 3.94989045143233784e-1, 6.06006527479721211e-16, 3.28317734584541611, 9.99999999999999822, 1.1282414671605333e-17, 1.1262194599794700e-17, 2.5070627600076907e-15, 2.246241229877970077e-15, 0.370768487417013987, 0.278450624942955571, 0.523838487672241171, 0.38862015251322781, 0.298156764974678579)\) where \(f(\vec{x}^*) = 32.6555929502463\).
(\(\mu, \lambda\))-CCMA-ES
for constrained black-box optimization

Daniel Wälchli
Computational Science & Engineering Laboratory ETH Zürich