(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

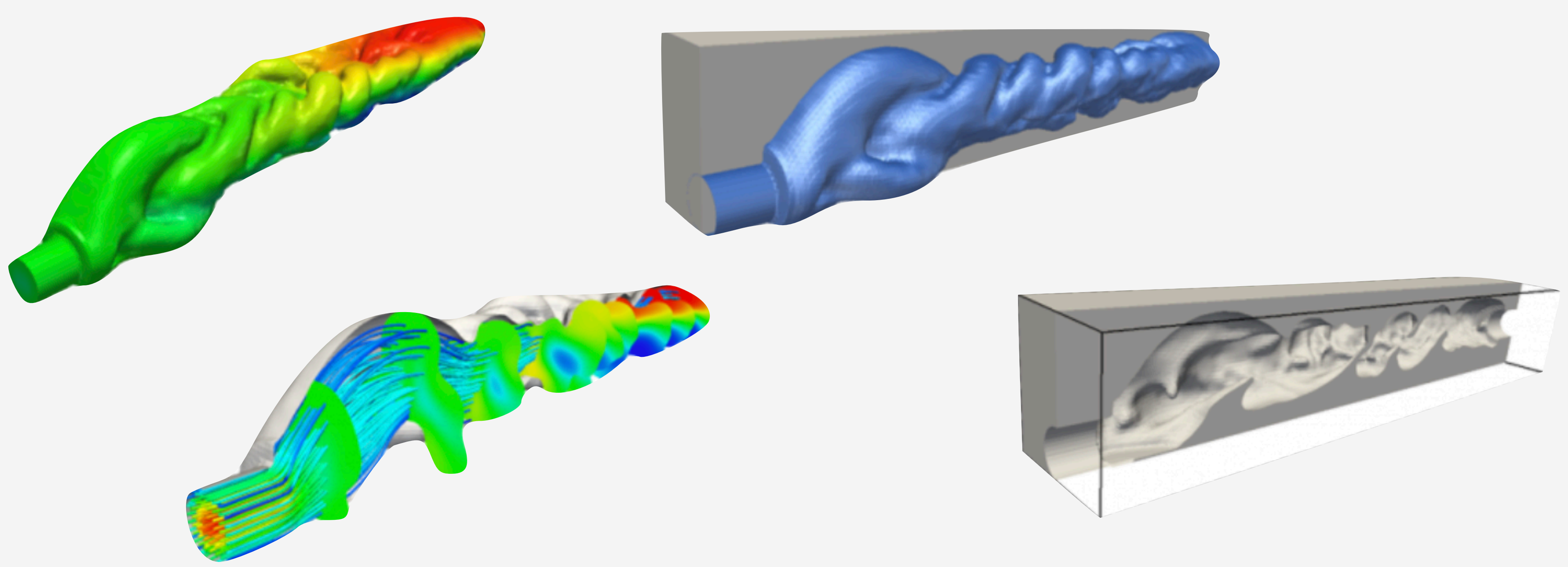
$$h_j(x) \leq 0, \quad j \in \{1, \ldots, m\}$$

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 $h_j(x)$

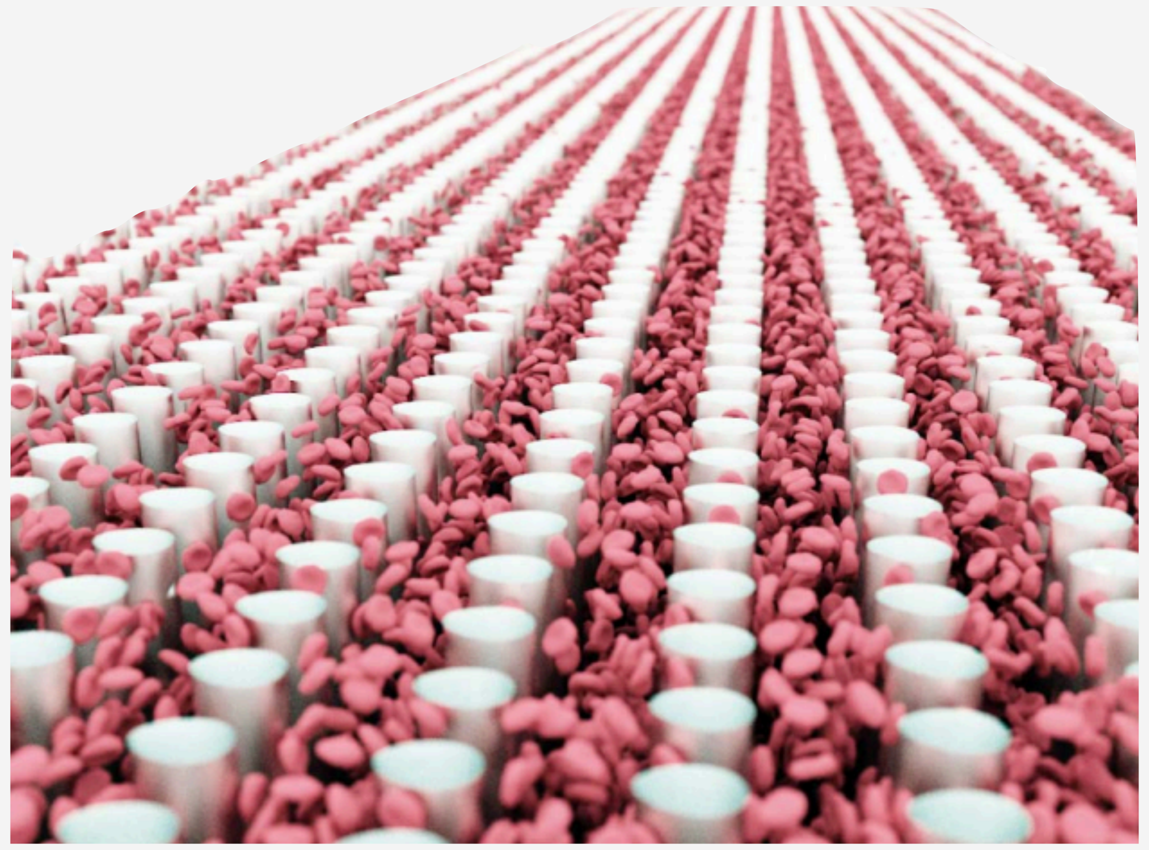
Motivation (cont.)

(constrained) optimization is ubiquitous, e.g:



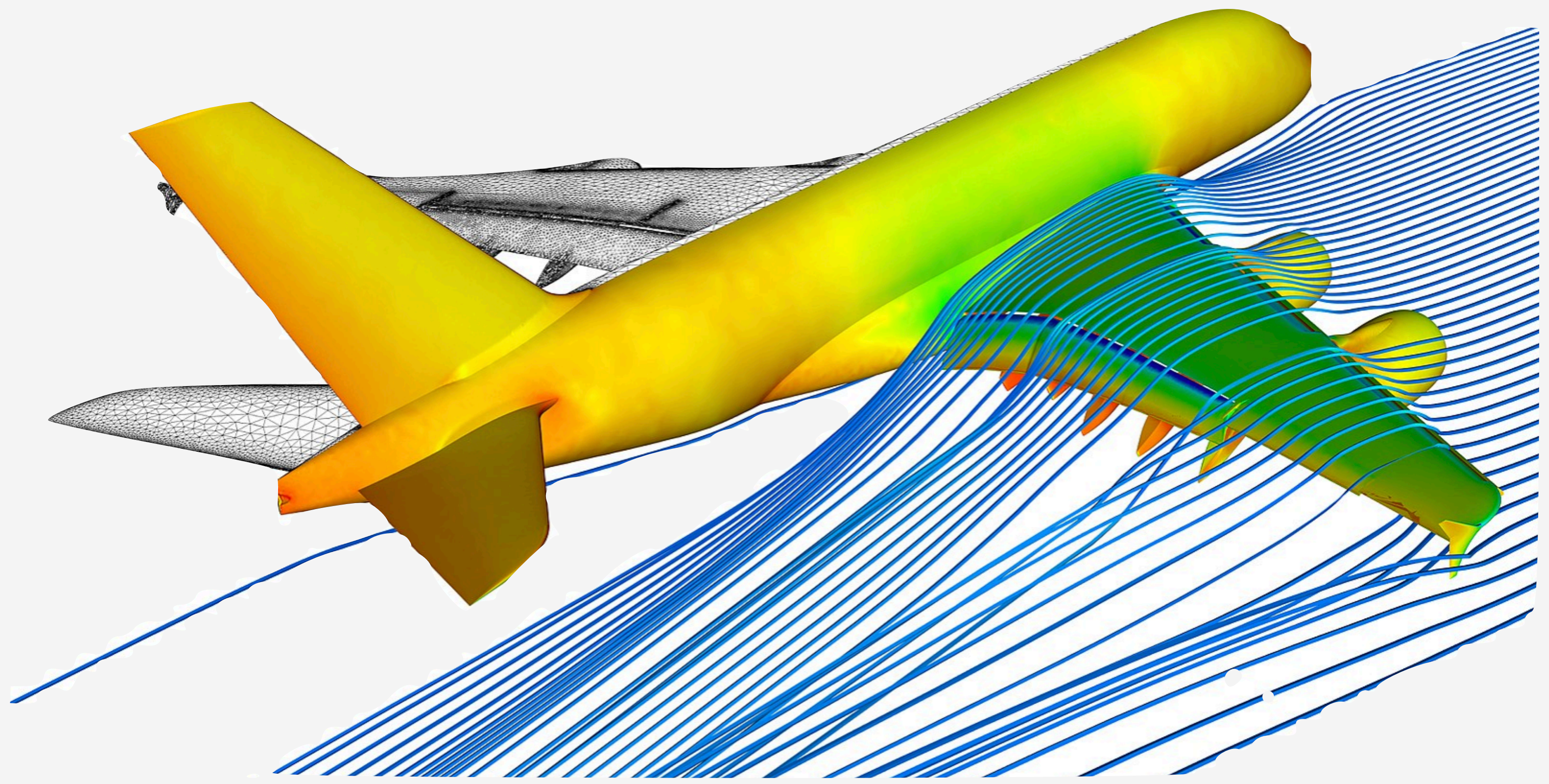
Mixing of Fluids

https://www.colorado.edu/center/aerospacemechanics/research/multi-physics-design-optimization



Cell Separation

https://www.cse-lab.ethz.ch/research/projects/#life-sciences

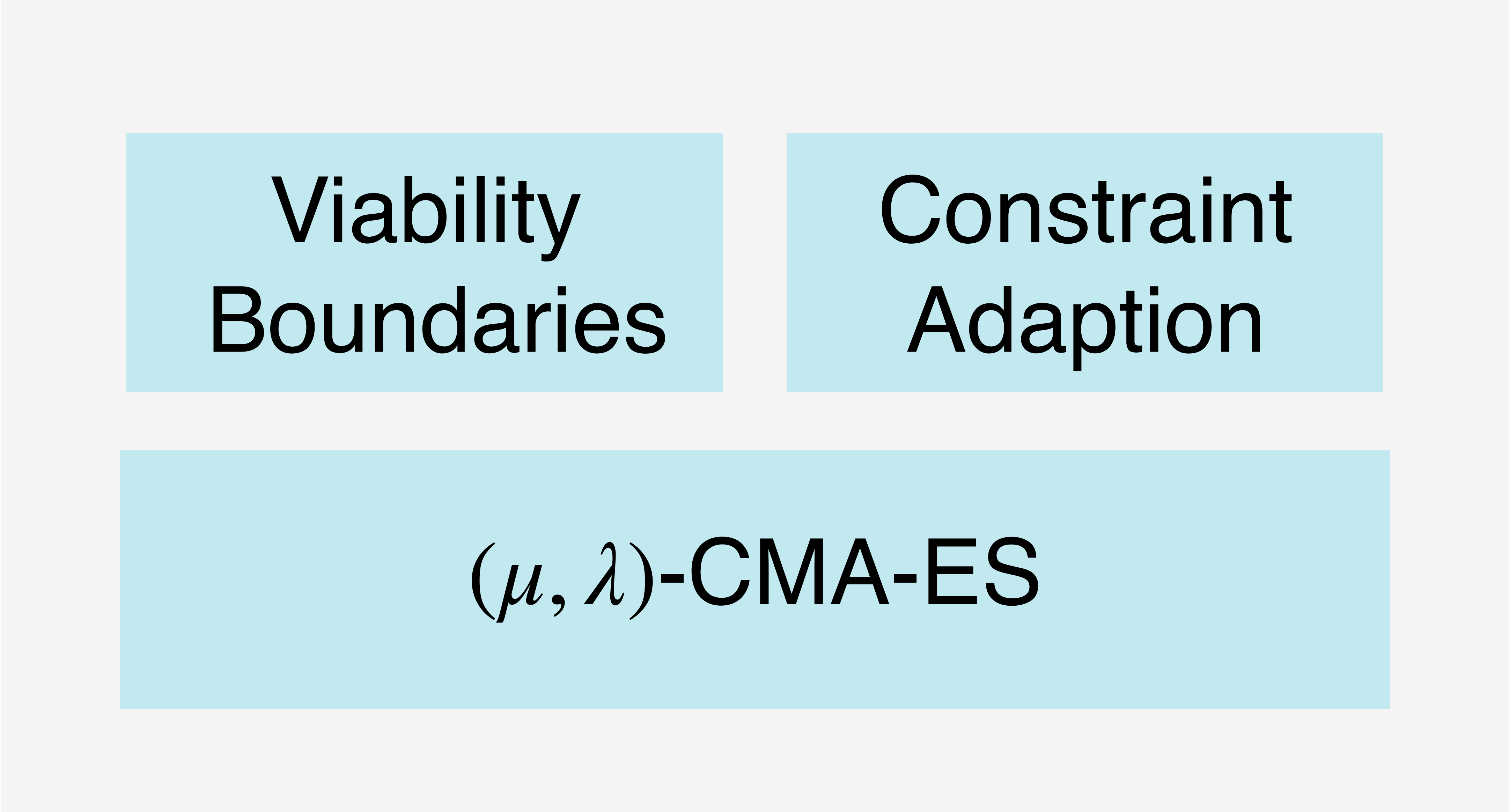


Drag Reduction

https://www.pinterest.ch/milanrohrer/engineering/

Background

(μ, λ)-CCMA-ES

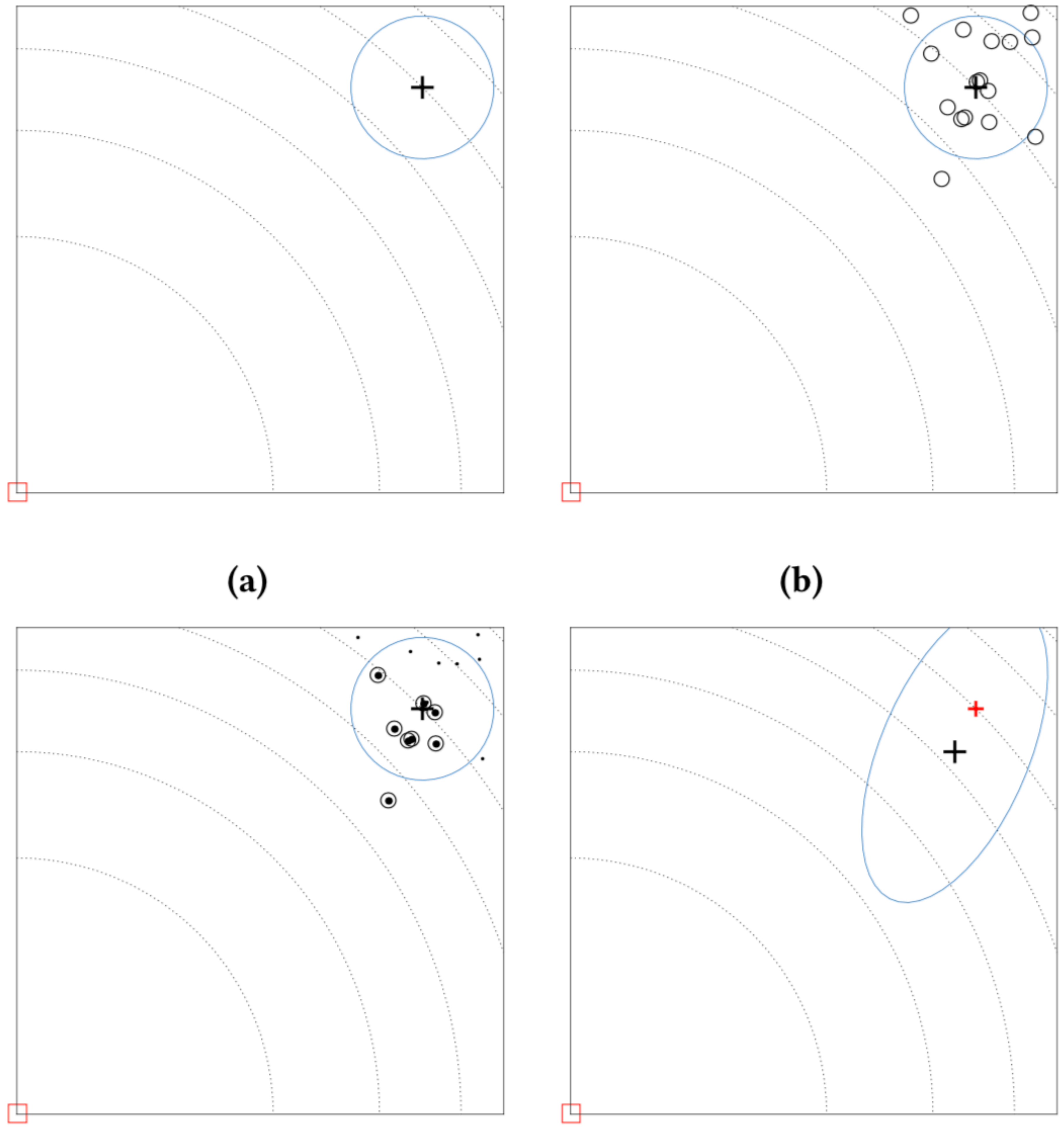


- **CMA-ES**: Covariance Matrix Adaption Evolution Strategy (CMA-ES) by Nikolaus Hansen [2016]. *https://arxiv.org/abs/1604.00772*
- **Viability Boundaries**: Viability Principles for Constrained Optimization Using a (1+1)-CMA-ES by Andreas Maesani and Dario Floreano [2014]. *Parallel Problem Solving from Nature – PPSN XIII. Springer International Publishing, Cham, 272–281*
- **Constraint Adaption:** A (1+1)-CMA-ES for Constrained Optimisation by Dirk V. Arnold and Nikolaus Hansen [2012]. *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO '12). ACM, New York, NY, USA, 297–304.*

(μ, λ)-CMA-ES

Algorithm 1 CMA-ES

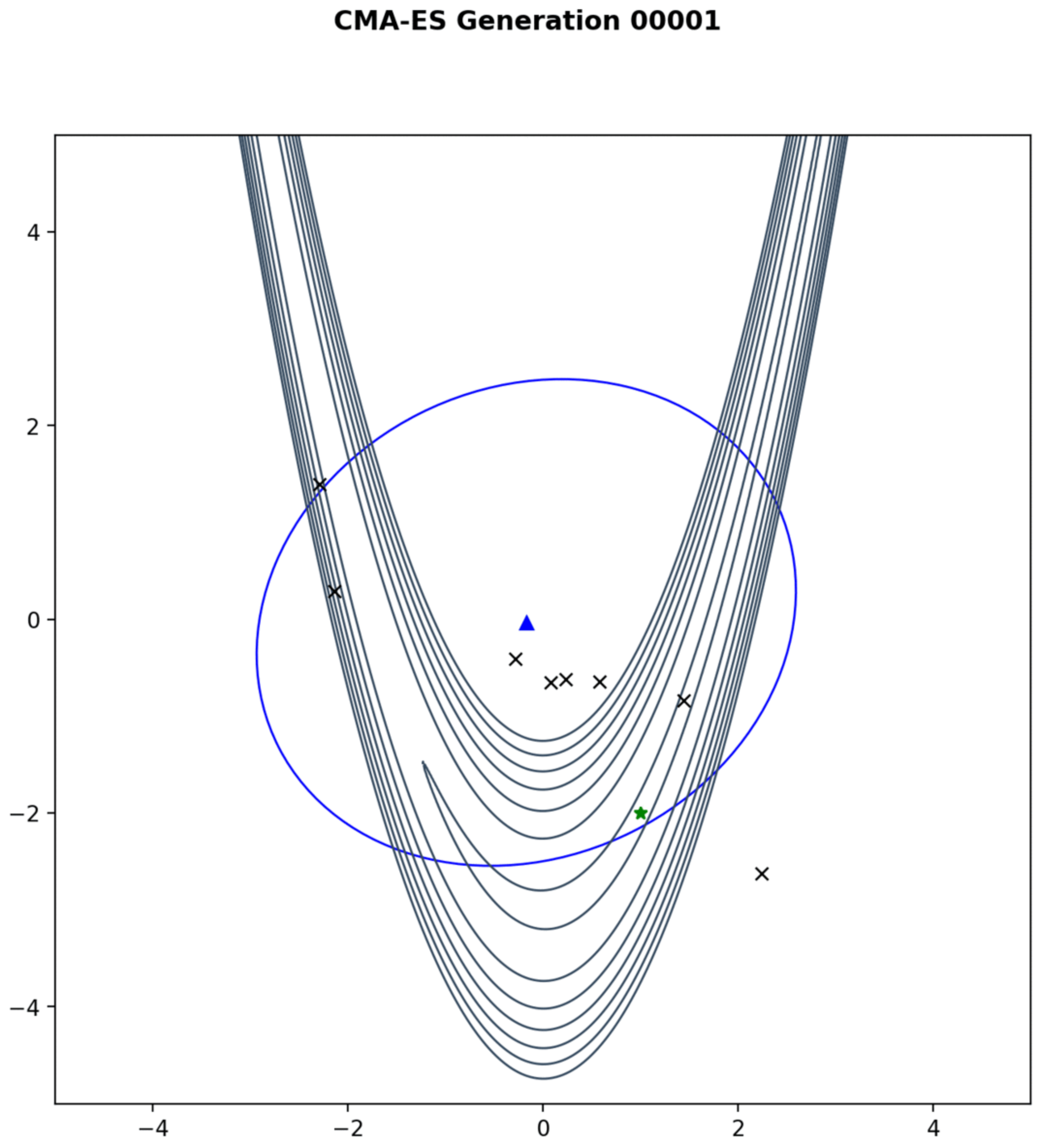
Initialize algorithm ▷ fig. 1a
while Termination criteria not met **do**
    Sampling $\mathbf{x_i} \sim \mathcal{N}(\mathbf{m}^{(g)}, \Sigma^{(g)})$ ▷ fig. 1b
    Evaluate individual fitness $f(\mathbf{x_i})$
    Selection and recombination ▷ fig. 1c
    Adaptation $\mathbf{m}^{(g+1)}$ and $\Sigma^{(g+1)}$ ▷ fig. 1d
end while
Return best ever found $\mathbf{x}^*$ and $f(\mathbf{x}^*)$



(a) (b) (c) (d)

- **Machine Learning**: CMA-ES for Hyperparameter Optimization of Deep Neural Networks by Ilya Loshchilov, Frank Hutter [2016]. *https://arxiv.org/abs/1604.07269*
- **Engineering**: Modular Approach for the Optimal Wind Turbine Micro Siting Problem through CMA-ES Algorithm by Silvio Rodrigues, Pavol Bauer, Jan Pierik [2013]. *GECCO '13 Companion. Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. Pages 1561-1568*
- **Image Processing:** The CMA-ES on Riemannian Manifolds to Reconstruct Shapes in 3-D Voxel Images by Sebastian Colutto, Florian Fruhauf, Matthias Fuchs, Otmar Scherzer [2009]. *IEEE Transactions on Evolutionary Computation. Vol. 14 Issue 2*

Example: (μ, λ)-CMA-ES



(μ, λ)-CCMA-ES

Algorithm 2 CCMA-ES

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

Viability Boundaries
&
Constraint Handling

discussion following slides

Constraint Handling

Algorithm 3 Constraint Handling in CCMA-ES

while Constraints violated **do**
  for $i = 1, \ldots, \lambda$ **do** ▷ for all offspring individuals
    for $j = 1, \ldots, m$ **do** ▷ for all constraints
      if $h_j(\boldsymbol{x}_i) > 0$ **then** ▷ if x_i violates constraint
        $\boldsymbol{v}_j \leftarrow (1 - c_v)\boldsymbol{v}_j + c_v \boldsymbol{y}_i$
        $C \leftarrow C - \frac{\beta}{\alpha_0(\boldsymbol{x}_i)} \frac{\boldsymbol{v}_j \boldsymbol{v}_j^\top}{\|\boldsymbol{v}_j\|^2}$
      end if
    end for
  end for
  for $i = 1, \ldots, \lambda$ **do** ▷ for all offspring individuals
    for $j = 1, \ldots, m$ **do** ▷ for all constraints
      if $h_j(\boldsymbol{x}_i) > 0$ **then** ▷ if x_i violates constraint
        Resample offspring $\boldsymbol{x}_i$
      end if
    end for
  end for
end while

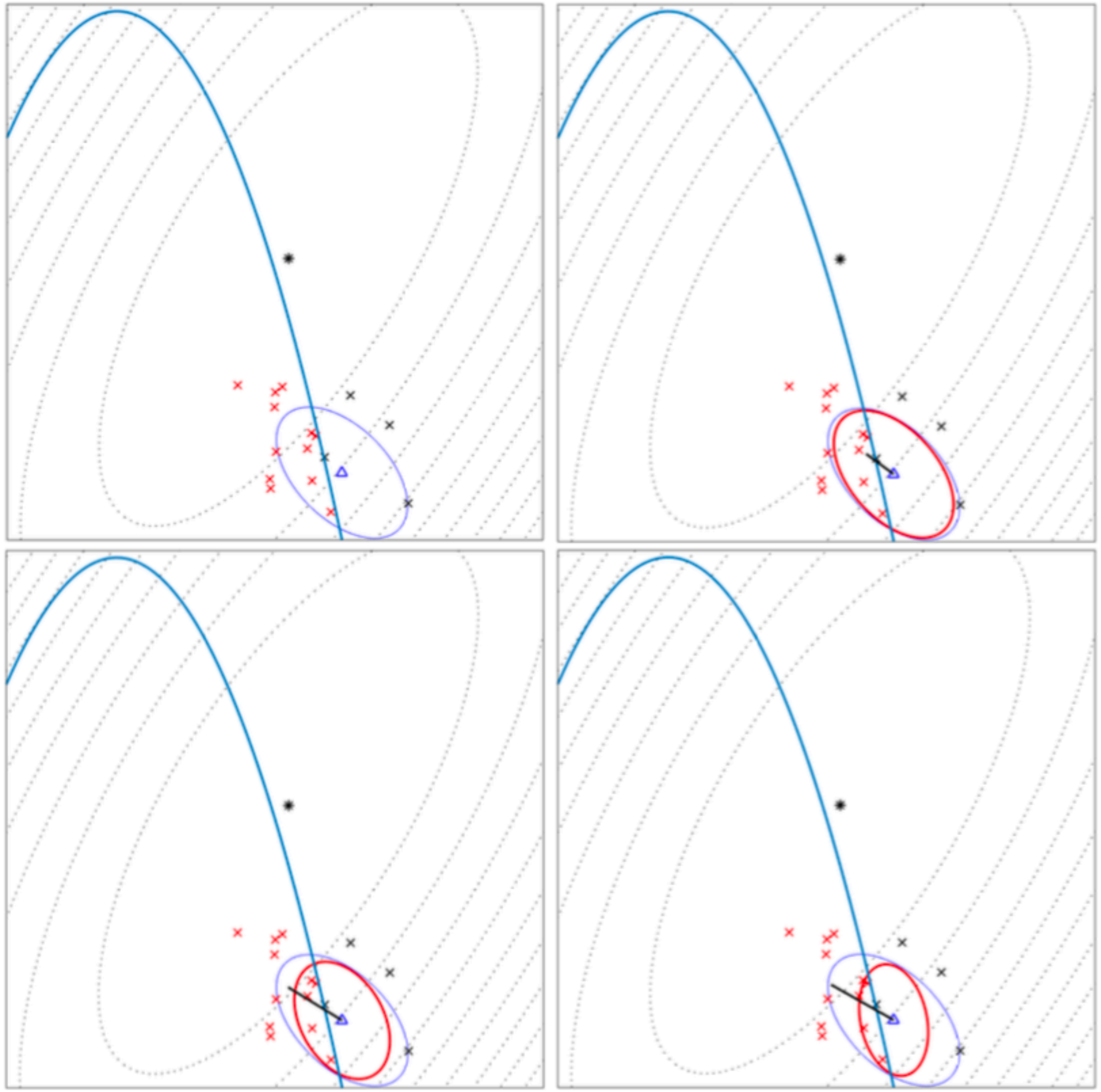
Constraint Normal Approximation

$$\boldsymbol{v}_j = (1 - c_v)\boldsymbol{v}_j + c_v \boldsymbol{y}_i$$

Covariance Matrix Correction

$$C = C - \frac{\beta}{\alpha_0(\boldsymbol{x})} \sum_{j=1}^{m} \alpha_j(\boldsymbol{x}) \frac{\boldsymbol{v}_j \boldsymbol{v}_j^\top}{\|\boldsymbol{v}_j\|^2}$$

Constraint Handling (cont.)



Constraint Normal Approximation

$$\boldsymbol{v}_j = (1 - c_v)\boldsymbol{v}_j + c_v \boldsymbol{y_i}$$

Covariance Matrix Correction

$$C = C - \frac{\beta}{\alpha_0(\boldsymbol{x})} \sum_{j=1}^{m} \alpha_j(\boldsymbol{x}) \frac{\boldsymbol{v}_j \boldsymbol{v}_j^\top}{\|\boldsymbol{v}_j\|^2}$$

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**:

$$\boldsymbol{b} = [\max\{0, h_1(\boldsymbol{x}_1), \ldots, h_1(\boldsymbol{x}_\lambda)\}, \ldots, \max\{0, h_m(\boldsymbol{x}_1), \ldots, h_m(\boldsymbol{x}_\lambda)\}]$$

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

$$\min\left\{b_i, h_i(\boldsymbol{x}_{\mathrm{c},i}) + \frac{b_i - h_i(\boldsymbol{x}_{\mathrm{c},i})}{2}\right\}, \text{ where}$$

$x_{\mathrm{c},i}$ denotes the sample closest to h_i

Viability Boundaries (cont.)

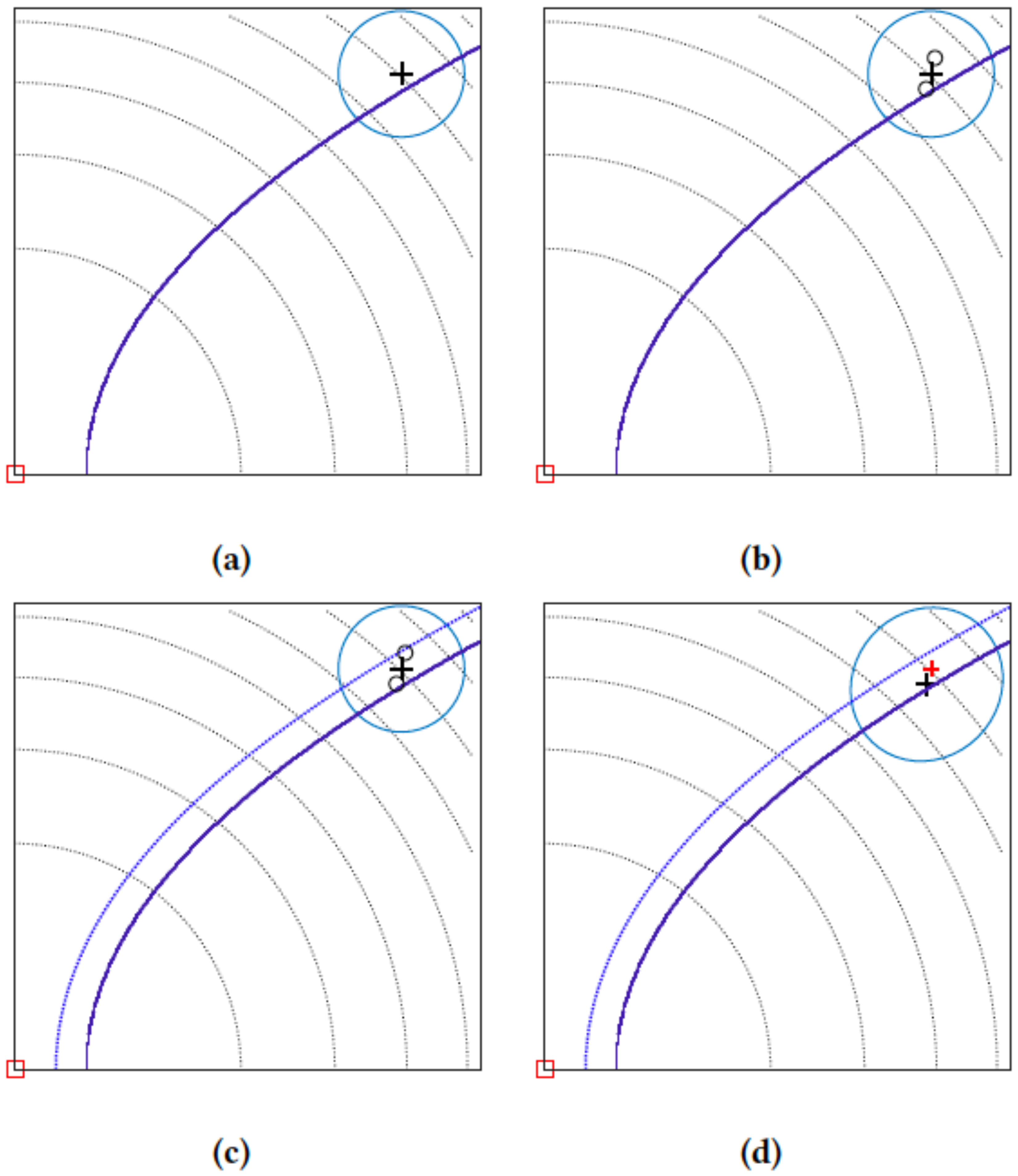
Discussion Viability Boundaries

fig. a: CCMA-ES initialized with mean $\mathbf{m}^{(0)}$ outside valid domain

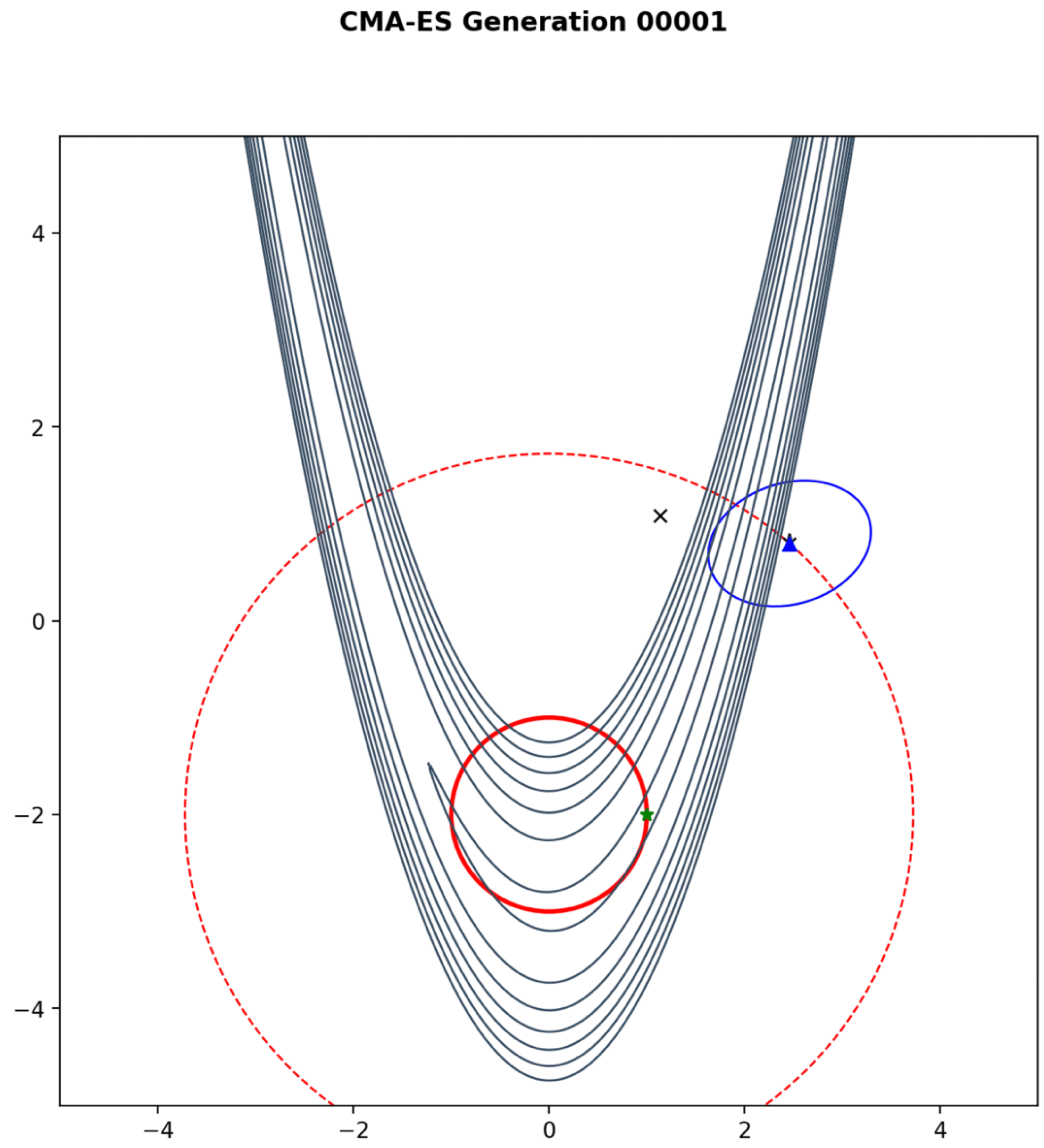
fig. b: Two samples created, both violating constraint h

fig. c: Relaxed boundary adapted to greatest constraint violation

fig. d: New proposal distribution calculated according to CMA-ES



Example: (μ, λ)-CCMA-ES



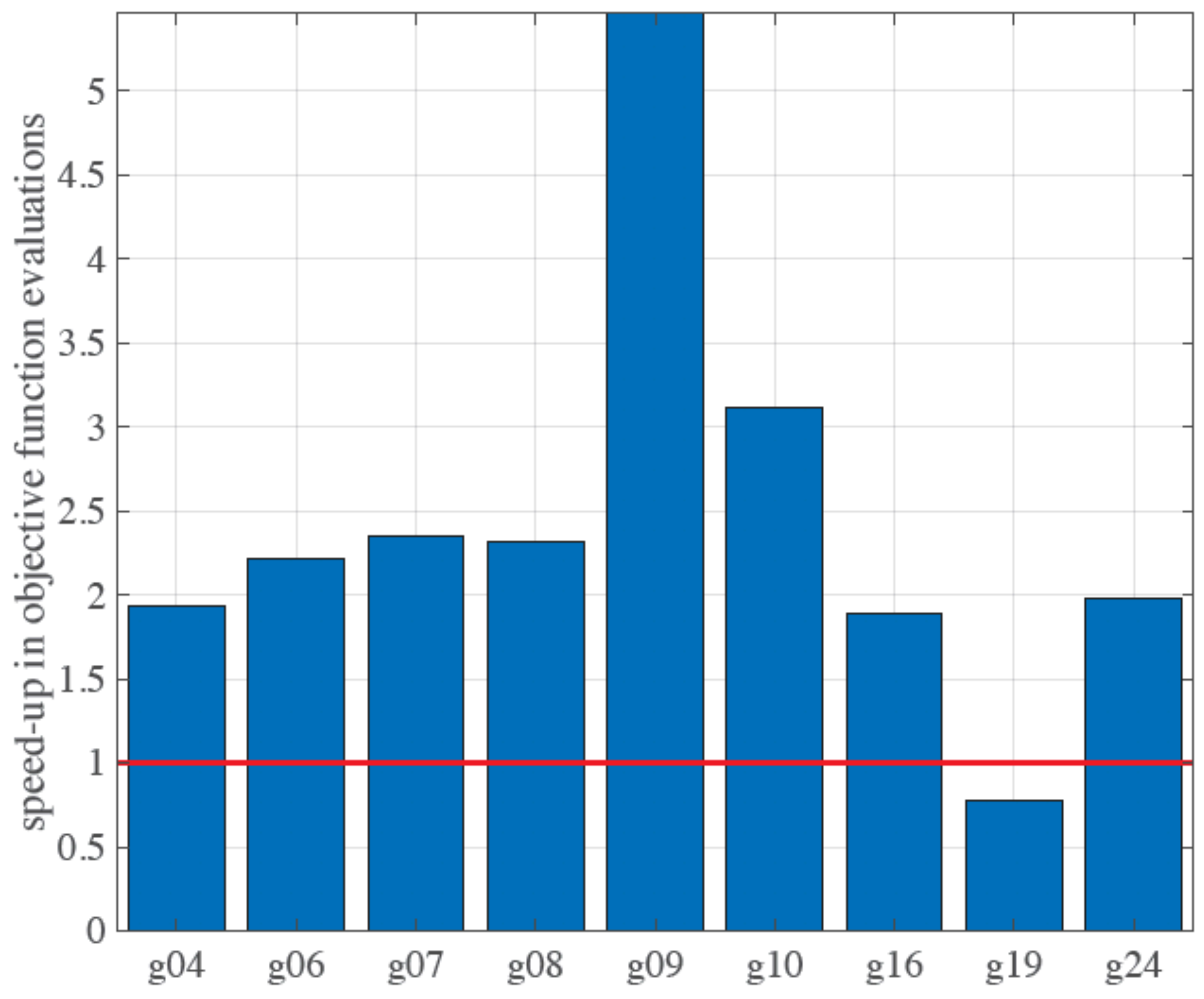
Evaluation

2006 CEC Test Problems

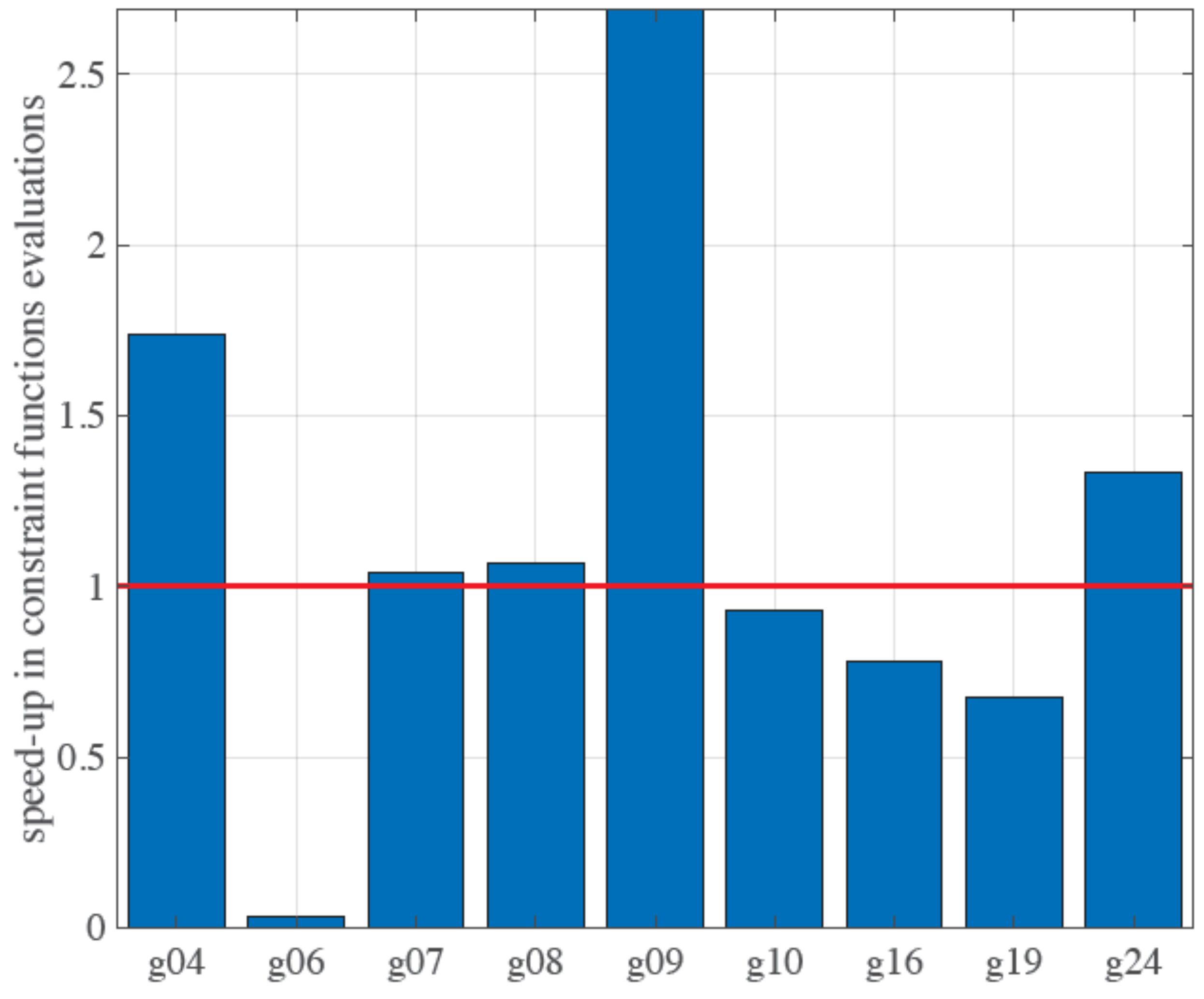
speed-up measured in terms of function evaluations

Baseline: mVIE

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



(a) Objective function evaluations



(b) Constraint function evaluations

- **mVIE**: Memetic Viability Evolution for Constrained Optimization [2016]. *IEEE Transactions on Evolutionary Computation 20, 1 (2016), 125-144*

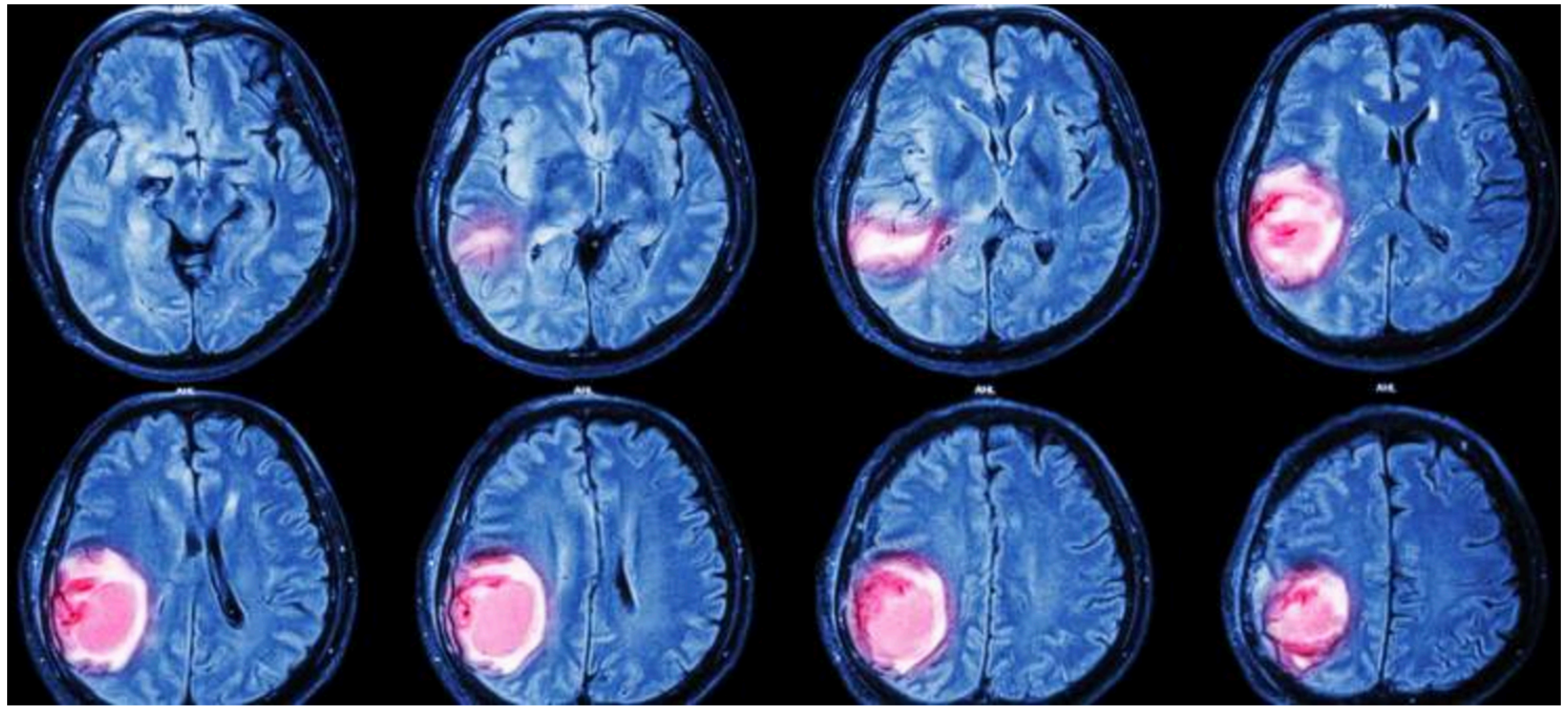
Evaluation (cont. I)

Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

$$\boldsymbol{x} = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q})$$

to reduce tumour size at t_{end} : $P^* = P + Q + Q_D$



http://www.iran-daily.com/News/206710.html

With respect to constraints:

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

$$h_{\max} = \max_t C(t) - v_{\max} \leq 0$$

$$h_c = \int_0^{t_{\text{end}}} C(t)dt - v_{\text{cum}} \leq 0$$

$$\begin{aligned}
\frac{dC}{dt} &= -\vartheta_1 C \\
\frac{dP}{dt} &= \vartheta_4 P\left(1 - \frac{P + Q + Q_D}{K}\right) + \vartheta_5 Q_D - \vartheta_3 P - \vartheta_1 \vartheta_2 C P \\
\frac{dQ}{dt} &= \vartheta_3 P - \vartheta_1 \vartheta_2 C Q \\
\frac{dQ_D}{dt} &= \vartheta_1 \vartheta_2 C Q - \vartheta_5 Q_D - \vartheta_6 Q_D ,
\end{aligned}$$

Evaluation (cont. II)

Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

$$\boldsymbol{x} = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q})$$

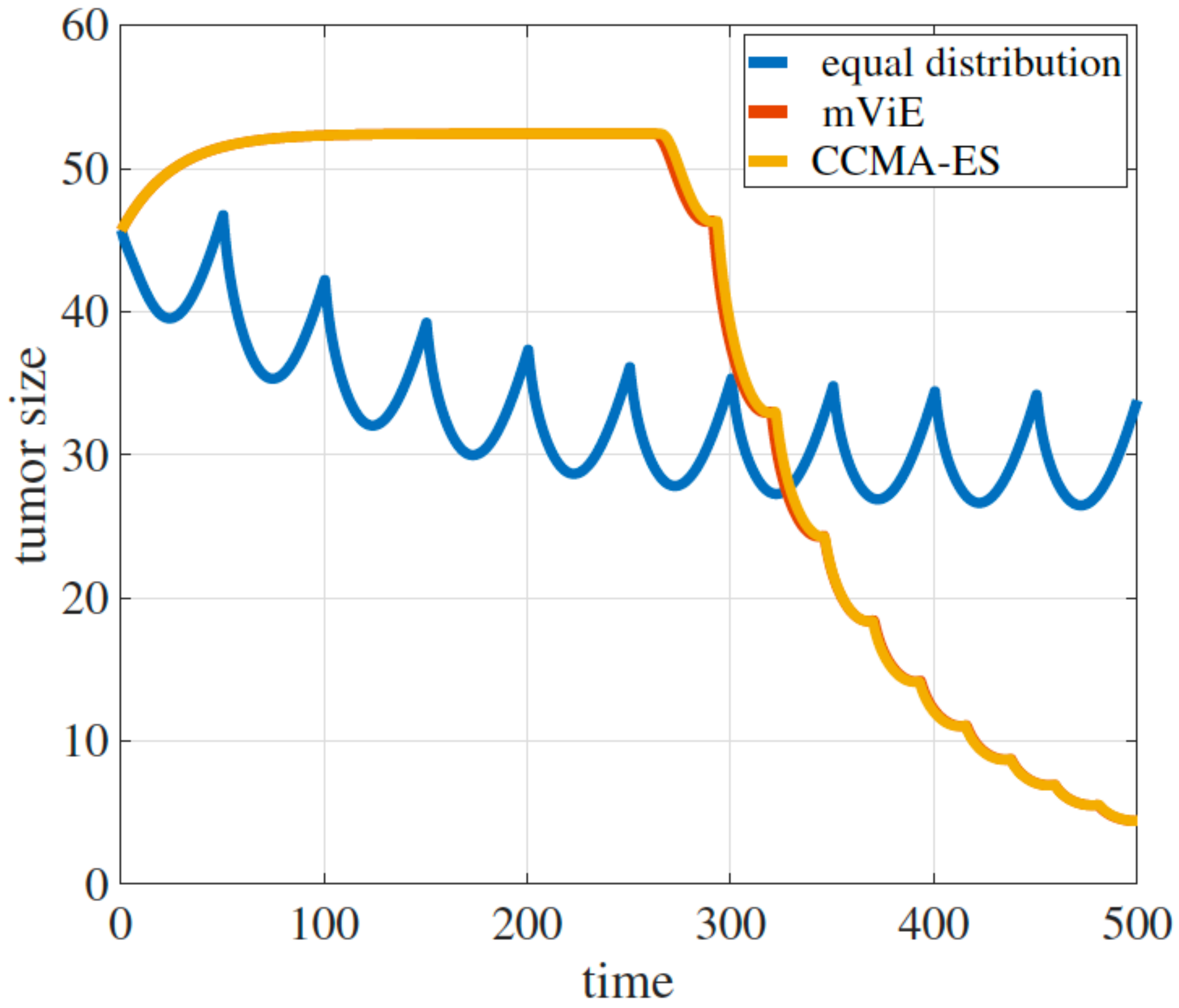
to reduce tumour size at t_{end} : $P^* = P + Q + Q_D$

With respect to constraints:

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

$$h_{\max} = \max_t C(t) - v_{\max} \leq 0$$

$$h_c = \int_0^{t_{\text{end}}} C(t)dt - v_{\text{cum}} \leq 0$$



$$\begin{aligned}
\frac{\mathrm{d}C}{\mathrm{d}t} &= -\vartheta_1 C \\
\frac{\mathrm{d}P}{\mathrm{d}t} &= \vartheta_4 P\left(1 - \frac{P + Q + Q_D}{K}\right) + \vartheta_5 Q_D - \vartheta_3 P - \vartheta_1 \vartheta_2 C P \\
\frac{\mathrm{d}Q}{\mathrm{d}t} &= \vartheta_3 P - \vartheta_1 \vartheta_2 C Q \\
\frac{\mathrm{d}Q_D}{\mathrm{d}t} &= \vartheta_1 \vartheta_2 C Q - \vartheta_5 Q_D - \vartheta_6 Q_D\,,
\end{aligned}$$

Evaluation (cont. III)

Pharmacodynamics for Tumor Growth

Find optimal treatment schedule

$$\boldsymbol{x} = (t_1, \ldots, t_{n_q}, a_1, \ldots, a_{n_q})$$

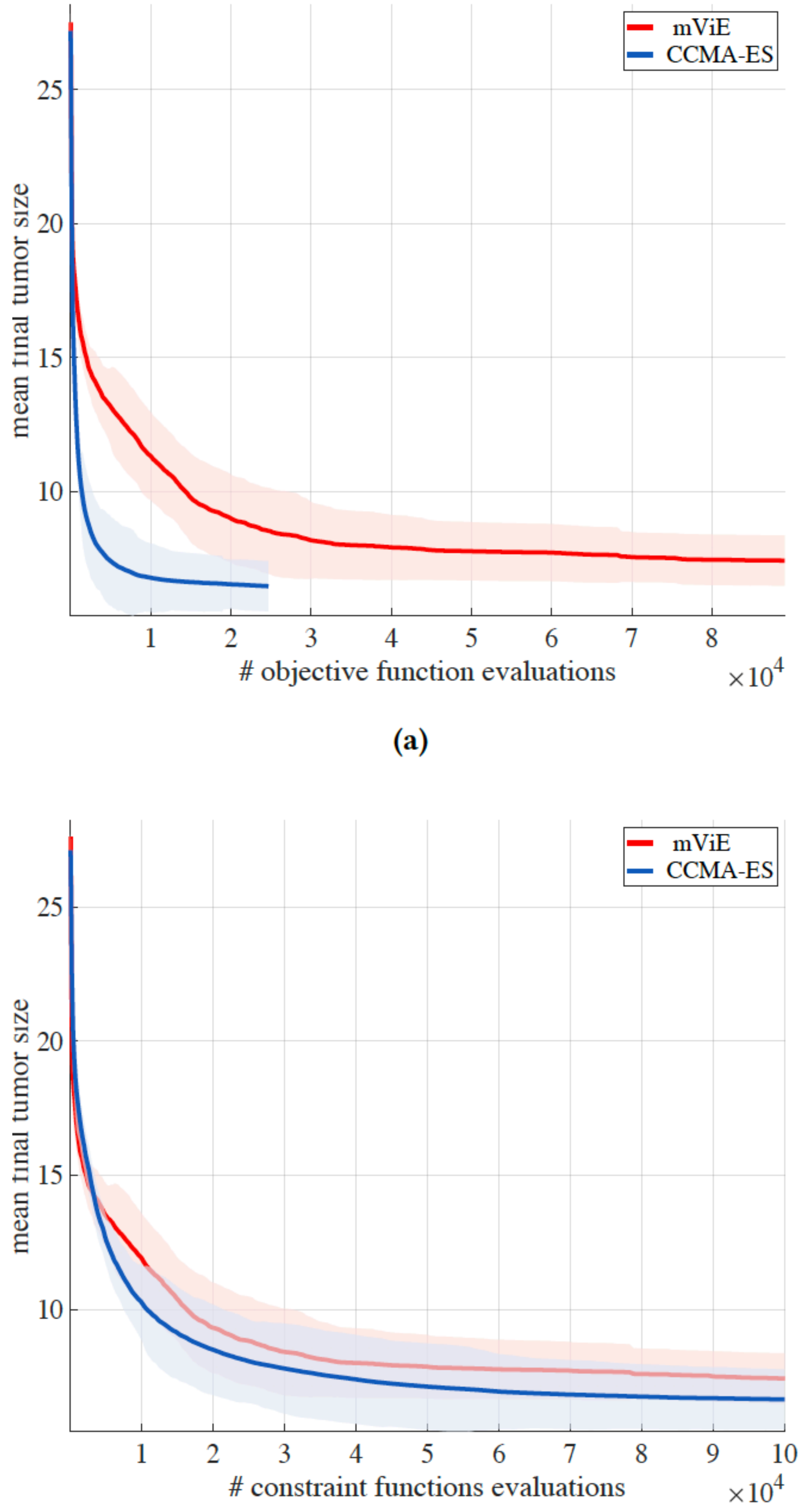
to reduce tumour size at t_{end} : $P^* = P + Q + Q_D$

With respect to constraints:

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

$$h_{\max} = \max_t C(t) - v_{\max} \leq 0$$

$$h_c = \int_0^{t_{\text{end}}} C(t)dt - v_{\text{cum}} \leq 0$$



(a)

(b)

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

Outlook

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

(μ, λ)-CMA-ES Updating Rule

Algorithm 1 CMA-ES

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

Evaluation. The objective function f is evaluated at the obtained individuals $\boldsymbol{x}_i$ and sorted $f(\boldsymbol{x}_{1:\lambda}) \leq f(\boldsymbol{x}_{2:\lambda}) \leq \cdots \leq f(\boldsymbol{x}_{\lambda:\lambda})$.

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

$$C^{g+1} = (1 - c_1 - c_\mu) C^g + c_1 \boldsymbol{p}_c^{g+1} \left(\boldsymbol{p}_c^{g+1}\right)^\top + c_\mu \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}^{g+1} \left(\boldsymbol{y}_{i:\lambda}^{g+1}\right)^\top$$

with $\boldsymbol{p}_c^{g+1} = (1 - c_c)\boldsymbol{p}_c^g + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \frac{\boldsymbol{m}^{g+1} - \boldsymbol{m}^g}{\sigma^g}$,

Examples 2006 CEC Test Problems

g09

Minimize [3]:

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to:

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

where $-10 \leq x_i \leq 10$ for $(i = 1, \ldots, 7)$. The optimum solution is $\vec{x}^* = (2.33049935147405174, 1.95137236847114592, -0.477541399510615805, 4.36572624923625874, -0.624486959100388983,$

g09: speed-up ~5.5 (best)

g19

Minimize [8]:

$$f(\vec{x}) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{(10+i)}x_{(10+j)} + 2\sum_{j=1}^{5} d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i \quad (39)$$

subject to:

$$g_j(\vec{x}) = -2\sum_{i=1}^{5} c_{ij}x_{(10+i)} - 3d_j x_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij}x_i \leq 0 \quad j = 1, \ldots, 5 \quad (40)$$

where $\vec{b} = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the remaining data is on Table 1. The bounds are $0 \leq x_i \leq 10$ $(i = 1, \ldots, 15)$. The best known solution is at $x^* = (1.66991341326291344e-17, 3.95378229282456509e-16, 3.94599045143233784, 1.06036597479721211e-16, 3.2831773458454161, 9.99999999999999822, 1.12829414671605333e-17, 1.2026194599794709e-17, 2.50706276000769697e-15, 2.24624122987970677e-15, 0.370764847417013987, 0.278456024942955571, 0.523838487672241171, 0.388620152510322781, 0.298156764974678579)$ where $f(x^*) = 32.6555929502463$.

g19: speed-up ~0.8 (worst)

(μ, λ)-CCMA-ES

for constrained black-box optimization

Daniel Wälchli

Computational Science & Engineering Laboratory ETH Zürich