Overview of Surrogate-model Versions of Covariance Matrix Adaptation Evolution Strategy

Zbyněk Pitra[1,2,3], Lukáš Bajer[1,4], Jakub Repický[1,4], Martin Holeňa[1]

[1]Institute of Computer Science, Czech Academy of Sciences
[2]Faculty of Nuclear Sciences and Physical Engineering
[3]National Institute of Mental Health
[4]Faculty of Mathematics and Physics, Charles University

Prague, Czech Republic

October 2, 2017

Contents

Continuous Black-box Optimization

- objective function evaluated **not analytically**, but:
  - empirically - via **measurements** (materials science, chemistry, ...)
  - through comprehensive numerical **simulations** (design of airplanes, crystal grows, ...)

Continuous Black-box Optimization

- objective function evaluated **not analytically**, but:
  - empirically - via **measurements** (materials science, chemistry, ...)
  - through comprehensive numerical **simulations** (design of airplanes, crystal grows, ...)
- **CMA-ES**
  - the state-of-the-art continuous black-box optimizer

Continuous Black-box Optimization

- objective function evaluated **not analytically**, but:
  - empirically - via **measurements** (materials science, chemistry, ...)
  - through comprehensive numerical **simulations** (design of airplanes, crystal grows, ...)
- **CMA-ES**
  - the state-of-the-art continuous black-box optimizer
- **expensive** scenario
  - limited number of function evaluations ($\#\mathrm{FEs}$) available

Continuous Black-box Optimization

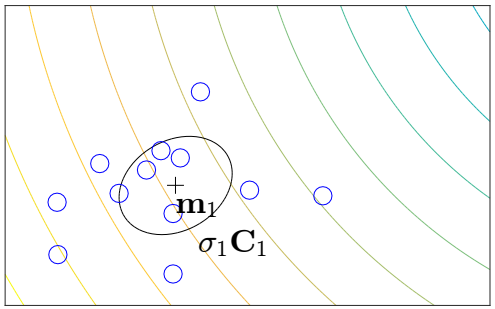
- objective function evaluated **not analytically**, but:
  - empirically - via **measurements** (materials science, chemistry, ...)
  - through comprehensive numerical **simulations** (design of airplanes, crystal grows, ...)
- **CMA-ES**
  - the state-of-the-art continuous black-box optimizer
- **expensive** scenario
  - limited number of function evaluations ($\#\mathrm{FEs}$) available
- **surrogate modelling**
  - saving on $\#\mathrm{FEs}$ by utilizing a model of the objective function

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n,\ \sigma \in \mathbb{R}_+,\ \mu, \lambda \in \mathbb{N},\ w_1, \ldots, w_\mu$
Initialize: $\boldsymbol{C} = \boldsymbol{I}\ (\ldots)$

while not terminate

1. $\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k,\ \boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}),\ k = 1, \ldots, \lambda$ {sampling}
2.
3.
4.
5.

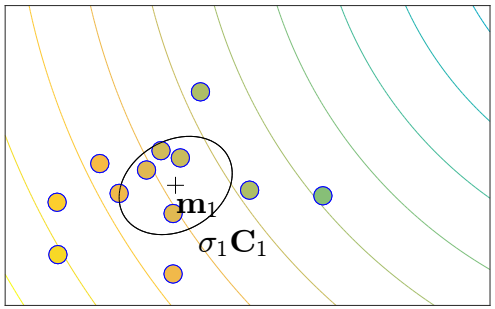


The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mu, \lambda \in \mathbb{N}$, $w_1, \ldots, w_\mu$
Initialize: $\boldsymbol{C} = \boldsymbol{I}$ $(\ldots)$

while not terminate

1. $\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k$, $\boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$, $k = 1, \ldots, \lambda$ {sampling}
2. evaluate $\boldsymbol{x}_k$ with the fitness
3.
4.
5.

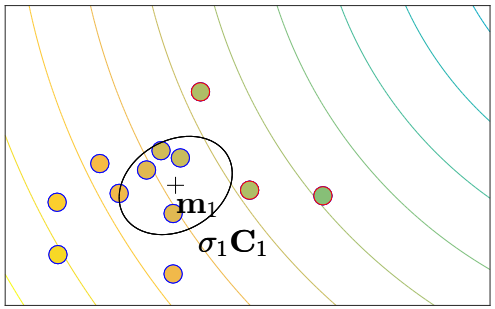


The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n,\ \sigma \in \mathbb{R}_+,\ \mu, \lambda \in \mathbb{N},\ w_1, \ldots, w_\mu$
Initialize: $\boldsymbol{C} = \boldsymbol{I}$ $(\ldots)$

while not terminate

1. $\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k,\ \boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}),\ k = 1, \ldots, \lambda$ {sampling}
2. evaluate $\boldsymbol{x}_k$ with the fitness
3. $\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i\, \boldsymbol{x}_{i:\lambda}$ {update mean}
4.
5.

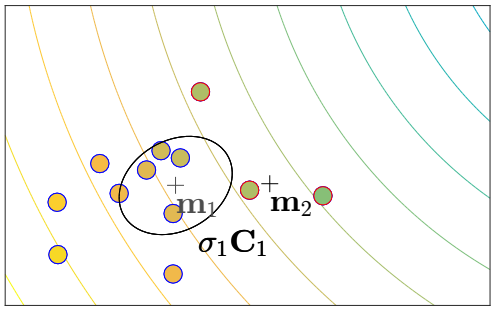


The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n,\ \sigma \in \mathbb{R}_+,\ \mu, \lambda \in \mathbb{N},\ w_1, \ldots, w_\mu$
Initialize: $\boldsymbol{C} = \boldsymbol{I}\ (\ldots)$

while not terminate

1. $\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k,\ \boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}),\ k = 1, \ldots, \lambda$ {sampling}
2. evaluate $\boldsymbol{x}_k$ with the fitness
3. $\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i\, \boldsymbol{x}_{i:\lambda}$ {update mean}
4.
5.

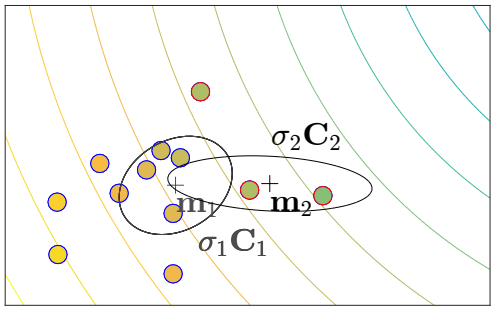


The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n,\ \sigma \in \mathbb{R}_+,\ \mu, \lambda \in \mathbb{N},\ w_1, \ldots, w_\mu$
Initialize: $\boldsymbol{C} = \boldsymbol{I}$ $(\ldots)$

while not terminate

1. $\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k,\ \boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}),\ k = 1, \ldots, \lambda$ {sampling}
2. evaluate $\boldsymbol{x}_k$ with the fitness
3. $\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i\, \boldsymbol{x}_{i:\lambda}$ {update mean}
4. update step-size σ
5. update $\boldsymbol{C}$



Gaussian Processes

Definition

A collection of random variables, any finite subset of which have a joint Gaussian distribution.

Gaussian Processes

Definition

A collection of random variables, any finite subset of which have a joint Gaussian distribution.

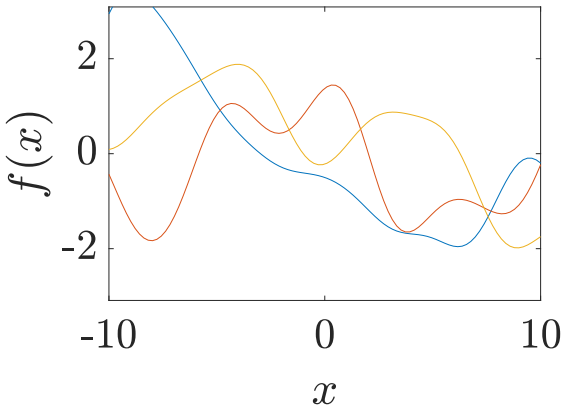
- specified by a **mean function** and a **covariance function**
- prediction in a point given as a **univariate** Gaussian (can represent **uncertainty**)

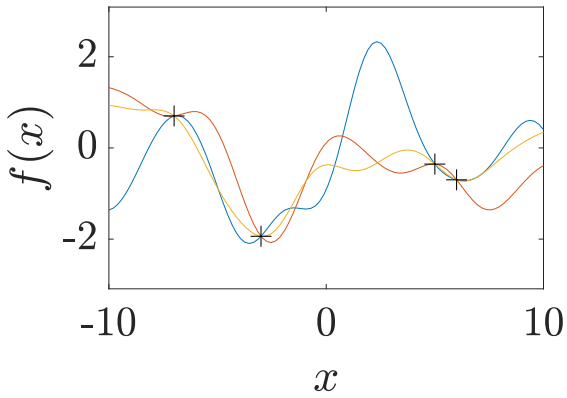
Gaussian Processes

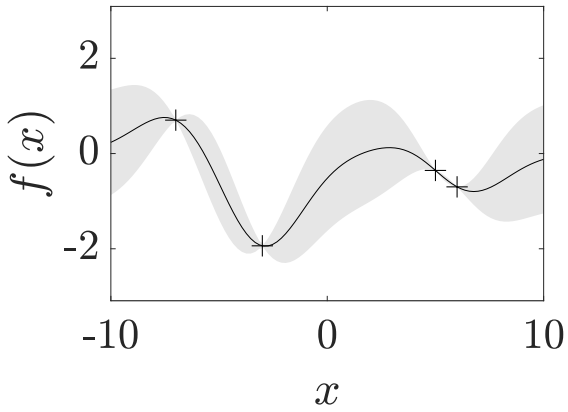
Definition

A collection of random variables, any finite subset of which have a joint Gaussian distribution.

- specified by a **mean function** and a **covariance function**
- prediction in a point given as a **univariate** Gaussian (can represent **uncertainty**)

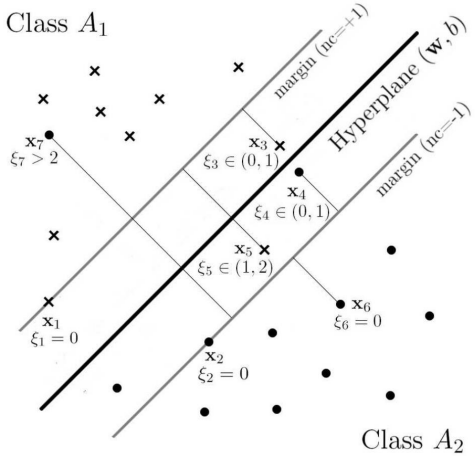


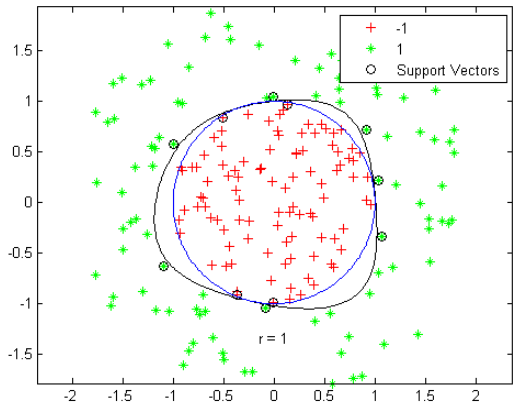




Ranking SVMs

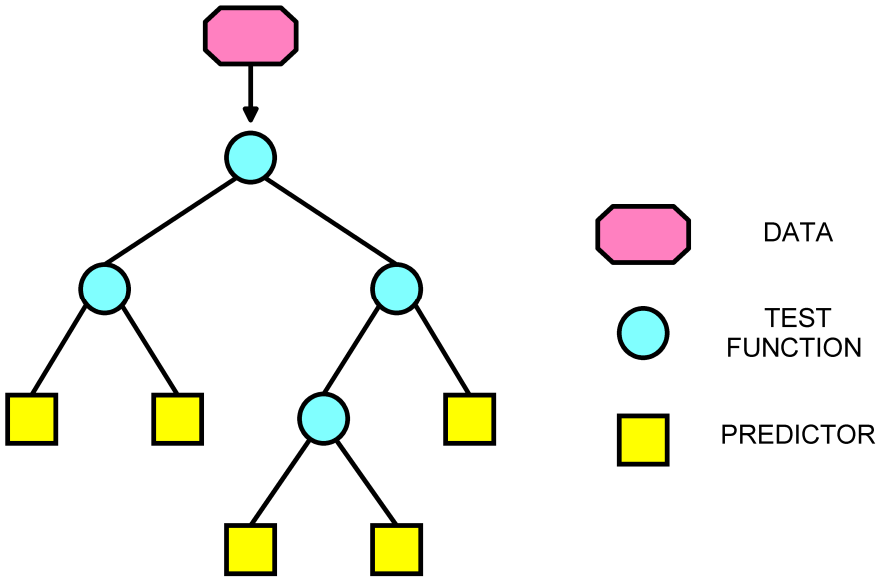
- **ordinal** version of Support Vector Machines
- separation through **maximizing margin** between levels
- **kernel trick** – mapping to separable space





Random Forests

- randomly trained ensembles of **decision trees**
- regression → regression trees
- superposition of many trees → **distribution estimator**



Metamodel-Assisted Evolution Strategy (MA-ES)

(Emmerich et al., 2002), (Ulmer et al., 2003)

- *Gaussian Process (GP)*
- μ parents are reproduced into $\lambda_{\mathrm{Pre}} > \lambda$ individuals
- GP model is used to **preselect** λ the most promising individuals
- various **preselection criteria** has been explored
- GP model is trained on N_{tr} of the most recently evaluated individuals

Local meta-model CMA-ES (lmm-CMA-ES)

(Kern et al., 2006), (Auger et al., 2013)

- **locally weighted regression**
  - individual *quadratic model* for each point $\mathbf{x}_i$
  - only k_{nn} nearest training points

Local meta-model CMA-ES (lmm-CMA-ES)

(Kern et al., 2006), (Auger et al., 2013)

- **locally weighted regression**
  - individual *quadratic model* for each point $\mathbf{x}_i$
  - only k_{nn} nearest training points
- **approximate ranking procedure**
  - repeated cycle of building models
  - termination 1: the ranking difference of two iterations $<$ treshold
  - termination 2: entire offspring has been originally-evaluated

Self-Adaptive Surrogate-Assisted CMA-ES (^{s}*ACM-ES)

(Loshchilov et al., 2012, 2013)

- *Ranking SVM*
- model is used for g_m generations
- original fitness f evaluates the population for one generation
- g_m adaptively changing according to the model error
- model parameters $\boldsymbol{\theta}$ optimized by the CMA-ES

Surrogate CMA-ES (S-CMA-ES)

(Bajer et al., 2015)

- *Gaussian processes* and *random forests*
- model is used for fixed number of generations
- original fitness f evaluates the population for one generation
- fixed model parameters $\boldsymbol{\theta}$

Doubly Trained Surrogate CMA-ES (DTS-CMA-ES)

(Pitra et al., 2016)

- *Gaussian processes*
- model $f_{\mathcal{M}1}$ predicts the whole distribution
- points selected using uncertainty criteria
- most promising points evaluated by the original fitness f
- rest of points evaluated by the re-trained model $f_{\mathcal{M}2}$

Experimental Setup

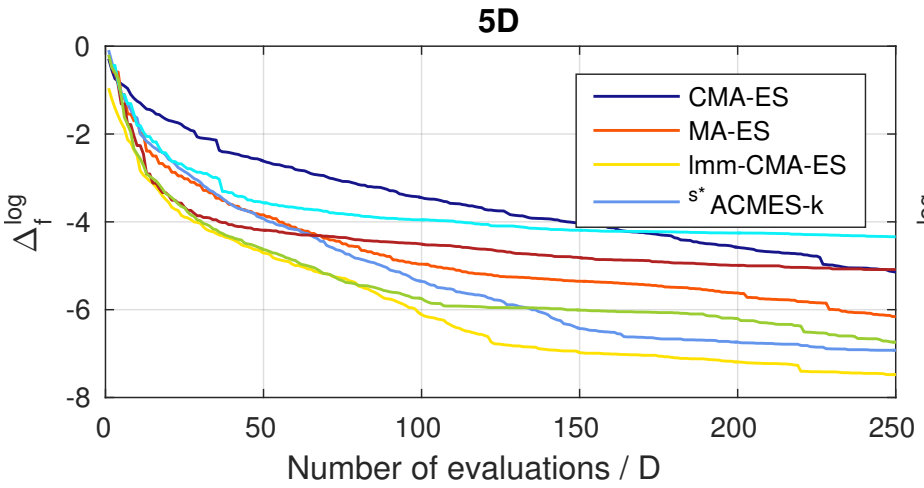
- Noiseless part of the BBOB testbed (24 functions)
- 2, 5, 10, 20D
- 15 instances
- 250 FE/D budget (expensive)
- target distance 10^{-8}

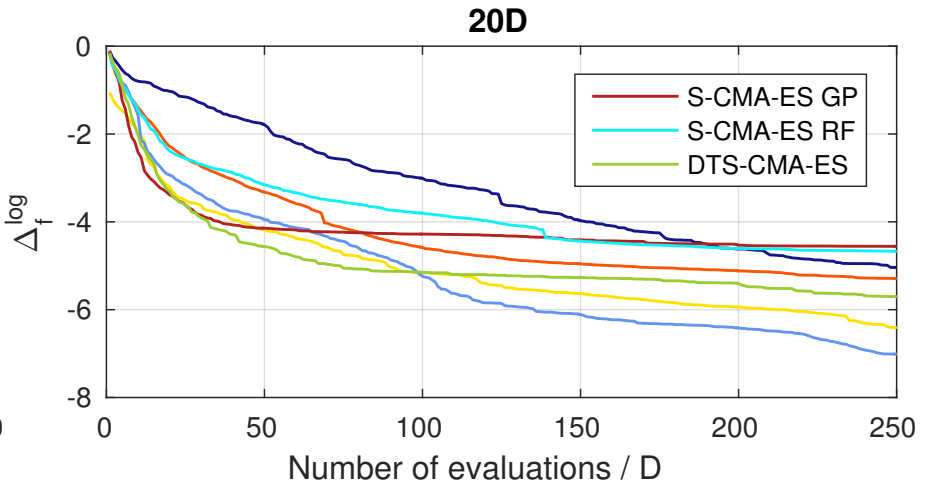
Algorithm Setup

Algorithms:

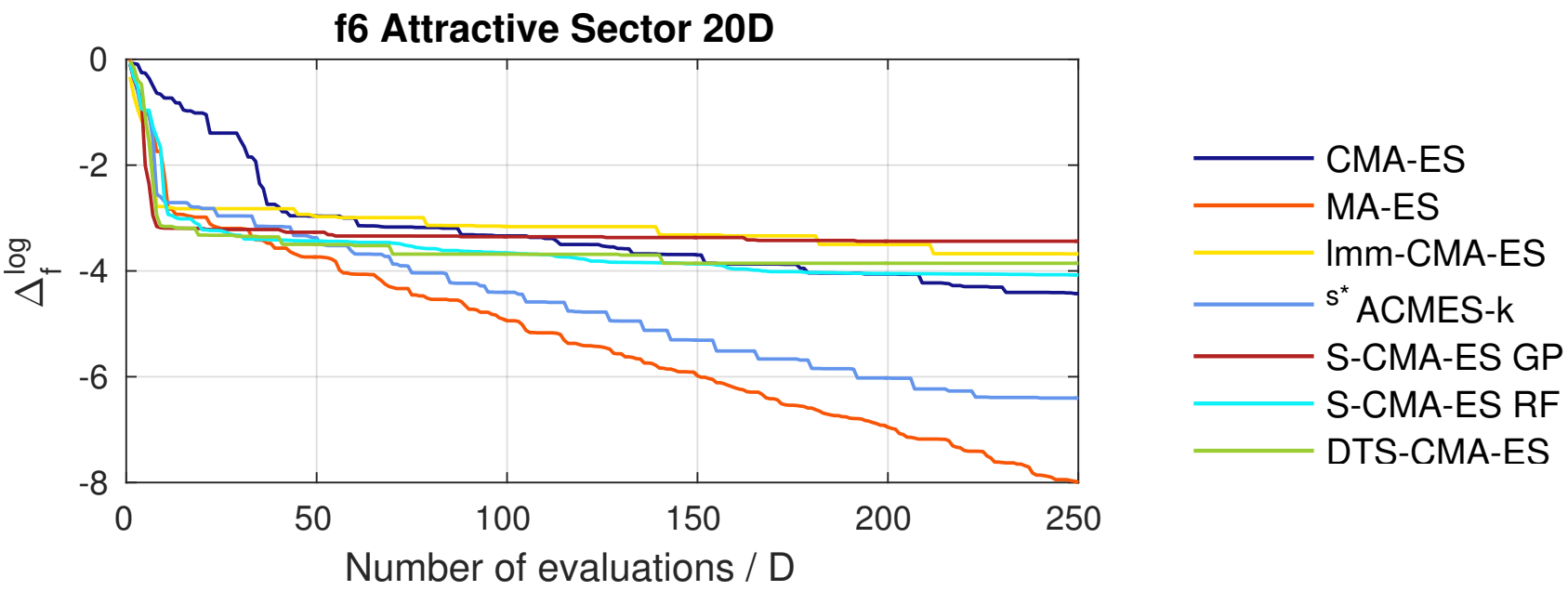
- CMA-ES (Hansen, 2006)
- MA-ES (Emmerich et al., 2002)
- lmm-CMA-ES (Auger et al., 2013)
- ^{s}*ACM-ES-k (Loshchilov et al., 2013)
- S-CMA-ES GP (Bajer et al., 2015)
- S-CMA-ES RF (Bajer et al., 2015)
- DTS-CMA-ES (Pitra et al., 2016)

BBOB results in 5 and 20D

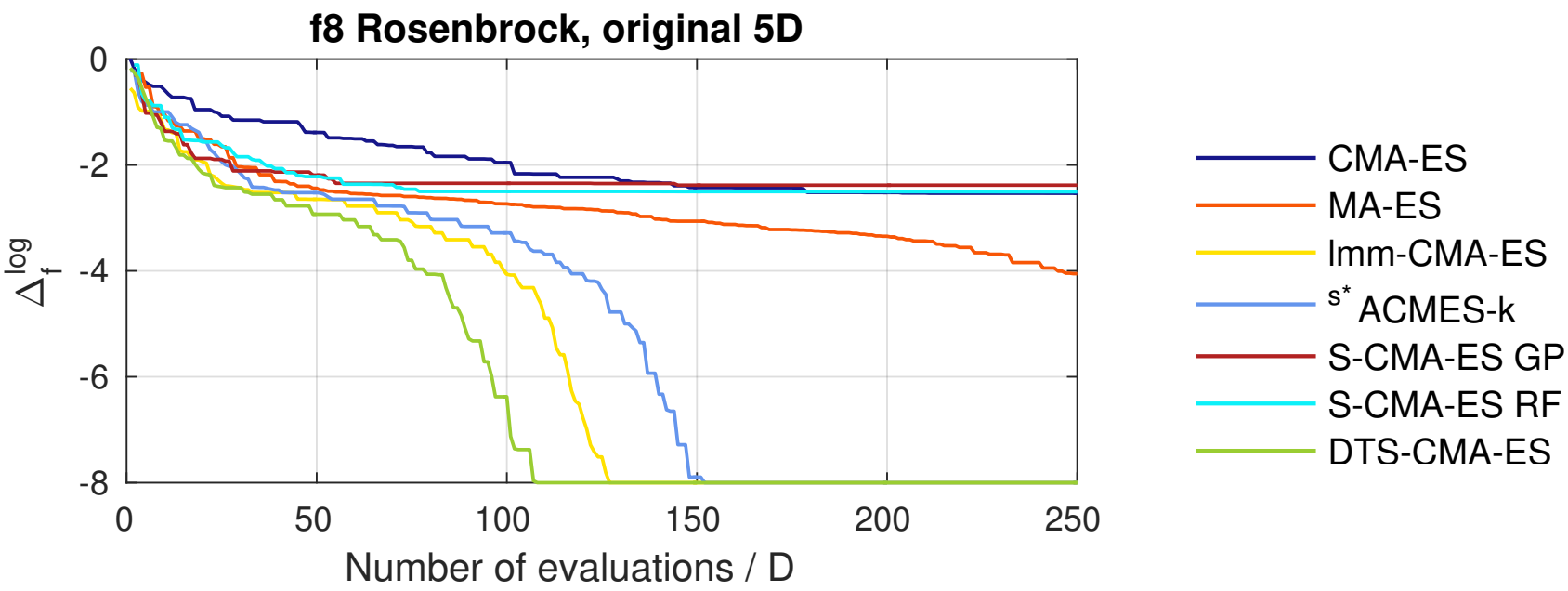




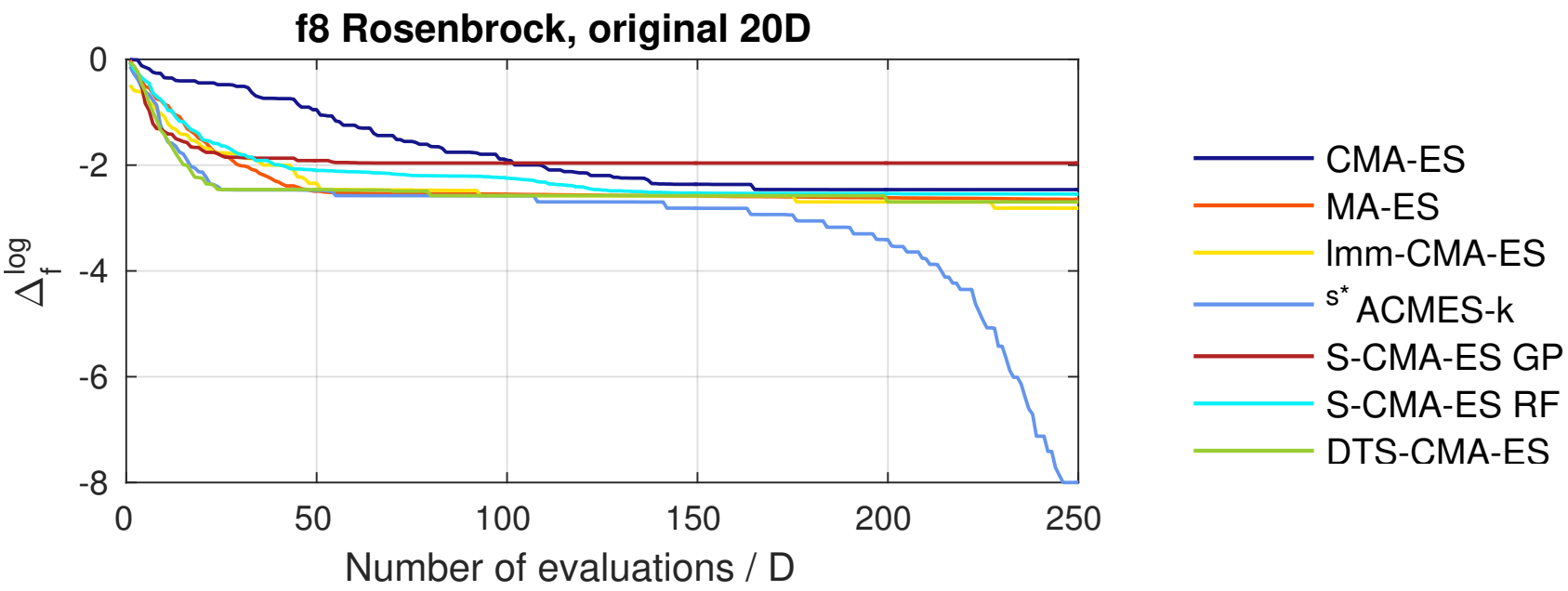
BBOB results on f6 in 20D



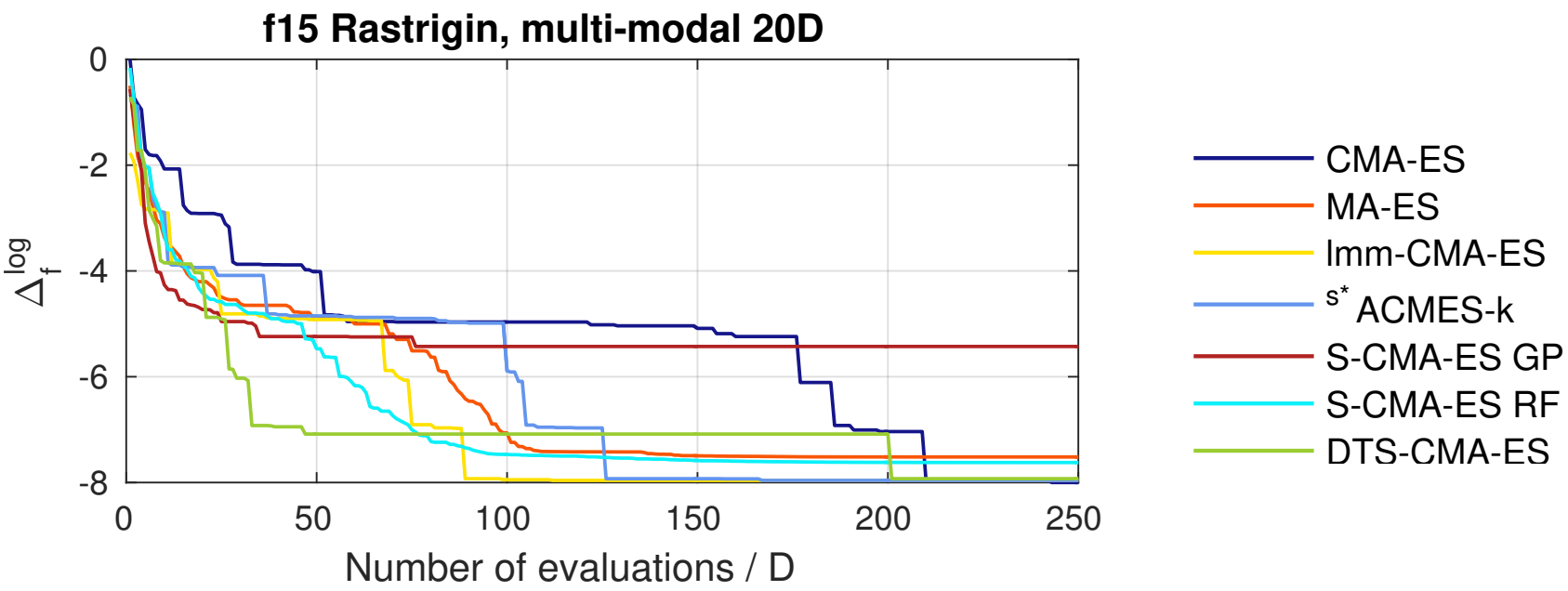
BBOB results on f8 in 5D



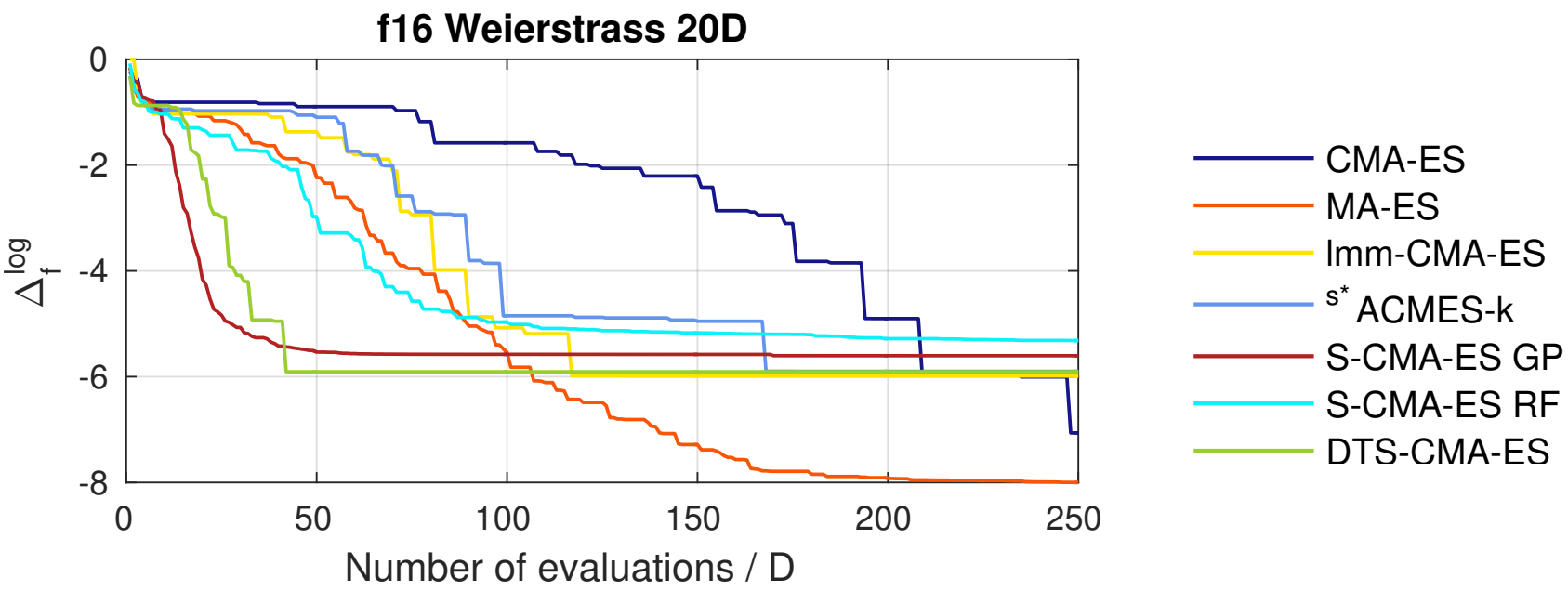
BBOB results on f8 in 20D



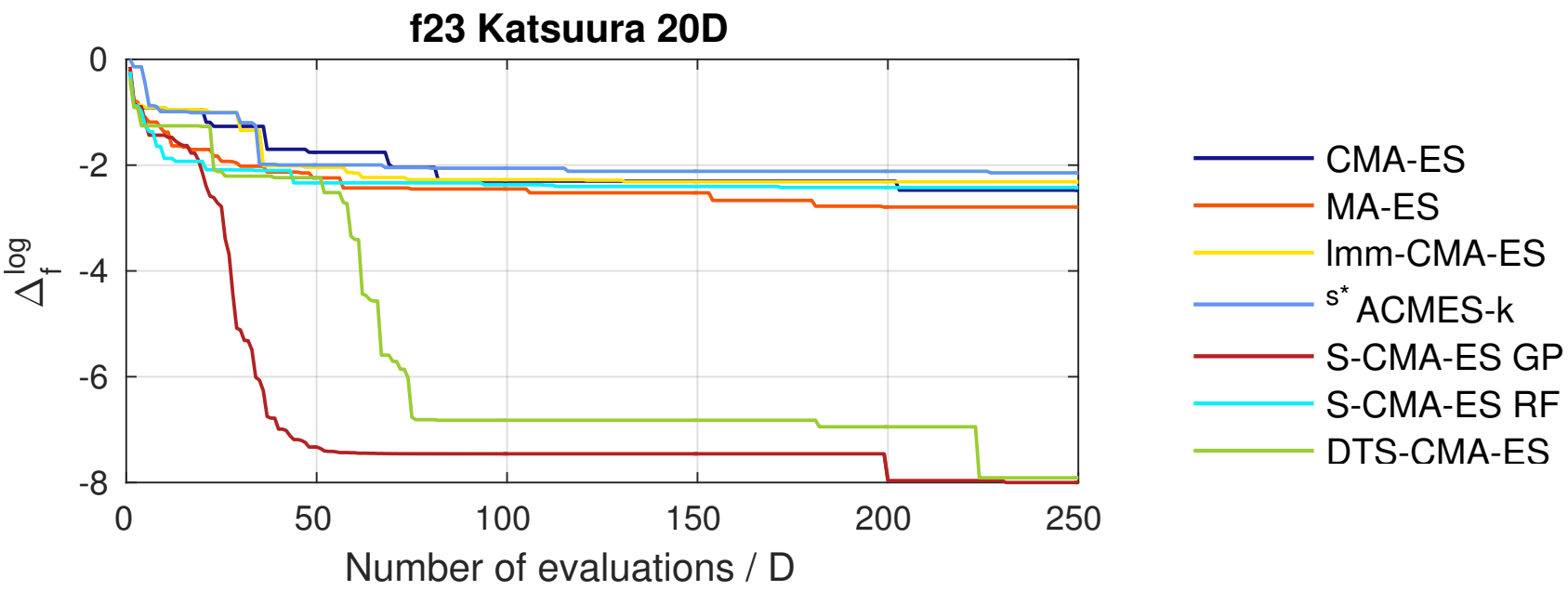
BBOB results on f15 in 20D



BBOB results on f16 in 20D



BBOB results on f23 in 20D



Comparison in 5D

$5D$	CMA-ES	MA-ES	lmm-CMA-ES	^{s}*ACM-ES-k	S-CMA-ES GP	S-CMA-ES RF	DTS-CMA-ES
CMA-ES	—	3	4	5	8	14	7
MA-ES	21	—	8	10	15	21	11
lmm-CMA-ES	20	16	—	16	21	22	13
^{s}*ACM-ES-k	19	14	8	—	17	21	9
S-CMA-ES GP	16	9	3	7	—	18	6
S-CMA-ES RF	10	3	2	3	6	—	4
DTS-CMA-ES	17	13	11	15	18	20	—

Table : The row algorithm achieving a significantly lower value of the objective function than the column algorithm is in red.

Comparison in 20D

$\mathbf{20}\boldsymbol{D}$	CMA-ES	MA-ES	lmm-CMA-ES	^{s}*ACM-ES-k	S-CMA-ES GP	S-CMA-ES RF	DTS-CMA-ES
CMA-ES	—	7	7	6	12	11	9
MA-ES	17	—	5	7	16	18	7
lmm-CMA-ES	17	19	—	10	21	20	19
^{s}*ACM-ES-k	18	17	14	—	20	20	18
S-CMA-ES GP	12	8	3	4	—	12	5
S-CMA-ES RF	13	6	4	4	12	—	3
DTS-CMA-ES	15	17	5	6	19	21	—

Table : The row algorithm achieving a significantly lower value of the objective function than the column algorithm is in red.

Conclusions

- All models significantly improve CMA-ES
- No best surrogate model or algorithm = no free lunch
- lmm-CMA-ES and ^{s}*ACM-ES
  - successful on broad spectrum of functions
  - follow CMA-ES invariances
- *Gaussian processes* most successful on multimodal functions

Thank you!

z.pitra@gmail.com bajeluk@gmail.com

j.repicky@gmail.com martin@cs.cas.cz