Benchmarking Gaussian Processes and Random Forests on the BBOB Noiseless Testbed

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

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

Prague, Czech Republic

July 2015

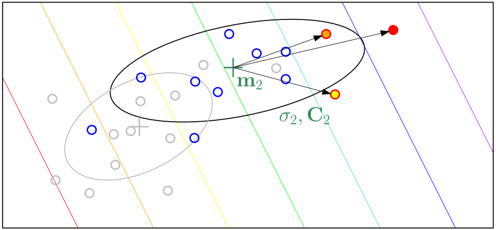
Contents

The CMA-ES

Input: $\mathbf{m} \in \mathbb{R}^n, \sigma \in \mathbb{R}_+, \lambda \in \mathbb{N}$
Initialize: $\mathbf{C} = \mathbf{I}$ (and several other parameters)
Set the weights $w_1, \ldots w_\lambda$ appropriately

while not terminate

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

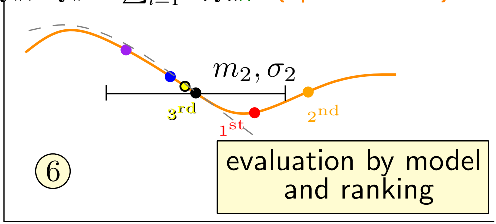


The Surrogate CMA-ES

Input: $\mathbf{m} \in \mathbb{R}^n, \sigma \in \mathbb{R}_+, \lambda \in \mathbb{N}$
Initialize: $\mathbf{C} = \mathbf{I}$ (and several other parameters)
Set the weights $w_1, \ldots w_\lambda$ appropriately

while not terminate

1. $\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i, \quad \mathbf{y}_i \sim N(\mathbf{0}, \mathbf{C}),$ for $i = 1, \ldots, \lambda$ {sampling}
2. evaluate $\mathbf{x}_i$ with the original fitness f & build a model $f_{\mathcal{M}}$ / evaluate $\mathbf{x}_i$ with the model $f_{\mathcal{M}}$
3. $\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \, \mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_w, \quad \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \, \mathbf{y}_{i:\lambda}$ {update mean}
4. update step-size σ
5. update $\mathbf{C}$



The Surrogate CMA-ES

Input: g (generation), $f_{\mathcal{M}}$ (model), $\mathcal{A}$ (archive), n_{REQ}, σ, λ, $\mathbf{m}$, $\mathbf{C}$

$\mathbf{x}_k \sim \mathcal{N}\left(\mathbf{m}, \sigma^2\mathbf{C}\right) \quad k = 1, \ldots, \lambda$ *{CMA-ES sampling}*

if g is original-evaluated **then**

  $y_k \leftarrow f(\mathbf{x}_k) \quad k = 1, \ldots, \lambda$ *{fitness evaluation}*

  $\mathcal{A} = \mathcal{A} \cup \{(\mathbf{x}_k, y_k)\}_{k=1}^{\lambda}$

  if $|\mathbf{X}| \geq n_{\text{REQ}}$ **then**

    $\mathbf{X} \leftarrow$ TransformToTheEigenvectorBasis($\mathbf{X}$, σ, $\mathbf{C}$)

    $f_{\mathcal{M}} \leftarrow$ trainModel$(\mathbf{X}, \mathbf{y})$

  end if

else

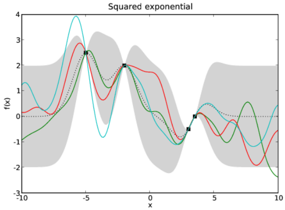
  $\mathbf{X} \leftarrow$ TransformToTheEigenvectorBasis($\mathbf{X}$, σ, $\mathbf{C}$)

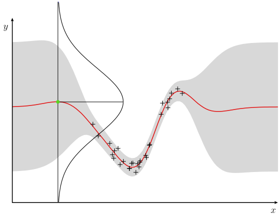
  $y_k \leftarrow f_{\mathcal{M}}(\mathbf{x}_k) \quad k = 1, \ldots, \lambda$ *{model evaluation}*

end if

Gaussian Process

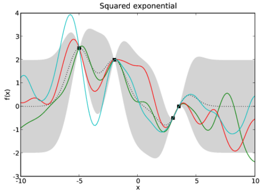
GP is a stochastic approximation method based on Gaussian distributions





GP can express **uncertainty** of the prediction in a new point $\mathbf{x}$: it gives a probability distribution of the output value

Gaussian Process



- given a set of N training points $\mathbf{X}_N = (\mathbf{x}_1 \ldots \mathbf{x}_N)^\top$, $\mathbf{x}_i \in \mathbb{R}^d$, and measured values $\mathbf{y}_N = (y_1, \ldots, y_N)^\top$ of a function f being approximated

$$y_i = f(\mathbf{x}_i), \quad i = 1, \ldots, N$$

GP considers vector of these function values as a sample from N-variate Gaussian distribution

$$\mathbf{y}_N \sim \mathrm{N}(\mathbf{0}, \mathbf{C}_N)$$

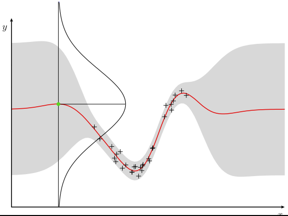
Gaussian Process prediction

Making predictions

Let $\mathbf{C}_{N+1}$ be extended covariance matrix – extended by entries belonging to an unseen point $(\mathbf{x}, y^*)$. Because $\mathbf{y}_N$ is known and the inverse $\mathbf{C}_{N+1}^{-1}$ can be expressed using inverse of the training covariance $\mathbf{C}_N{}^{-1}$,

the density in a new point marginalize to 1D Gaussian density

$$p(y^* \,|\, \mathbf{X}_{N+1}, \mathbf{y}_N) \;\propto\; \exp\left(-\frac{1}{2}\frac{(y^* - \hat{y}_{N+1})^2}{s^2_{y_{N+1}}}\right)$$

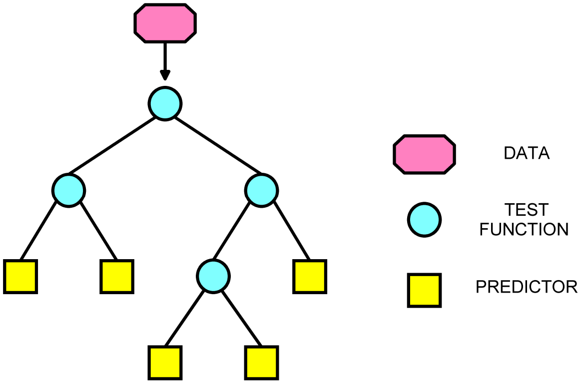


with the mean and variance given by

$$\begin{aligned} \hat{y}_{N+1} &= \mathbf{k}^\top \mathbf{C}_N{}^{-1} \mathbf{y}_N, \\ s^2_{y_{N+1}} &= \kappa - \mathbf{k}^\top \mathbf{C}_N{}^{-1} \mathbf{k}. \end{aligned}$$

Decision tree

A **decision tree** is a tree where each split node stores a test function to be applied to the incoming data and each leaf stores a predictor.



Decision tree
Advantages and disadvantages

Advantages:

- Relatively fast
- Easy to interpret
- Adaptive — structure and parameters learned from training data

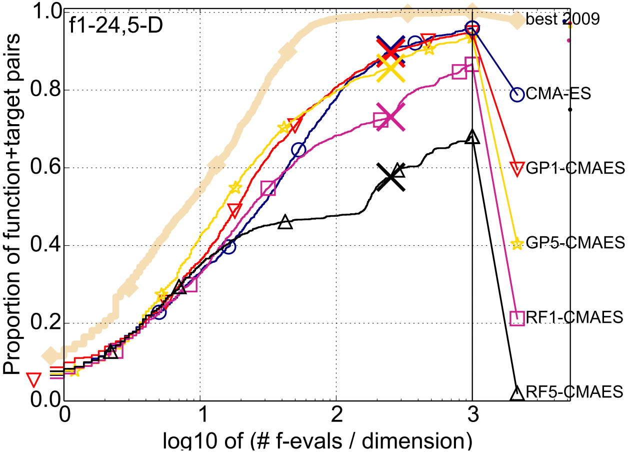
Disadvantages:

- Sharp decision boundaries
- Not the best predictive accuracy

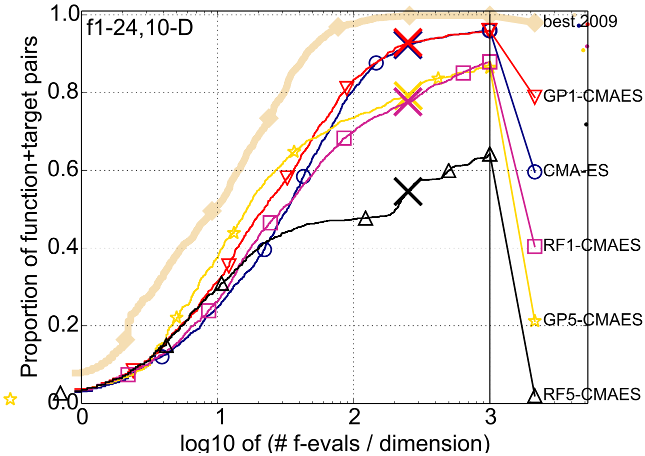
Random forests

- A collection of randomly trained decision trees
- Overall prediction determined by averaging
- All advantages of decision trees

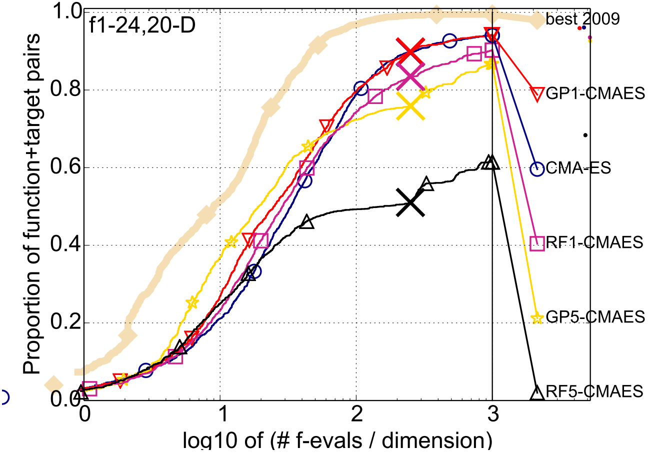
Experimental results on BBOB (5 D)



Experimental results on BBOB (10 D)

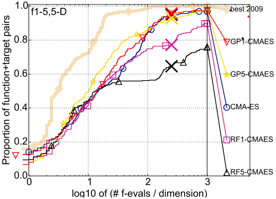


Experimental results on BBOB (20 D)

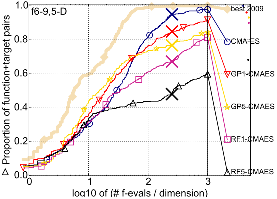


ECDF results on the whole BBOB (5 D)

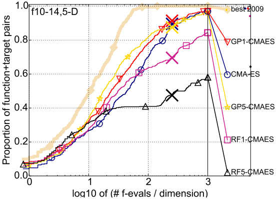
separable



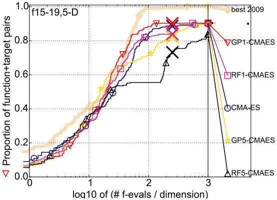
moderate



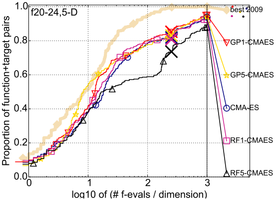
ill-conditional



multi-modal

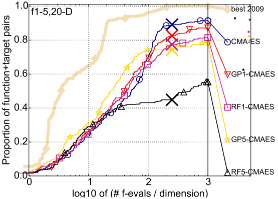


weakly structured multi-modal ill-conditional

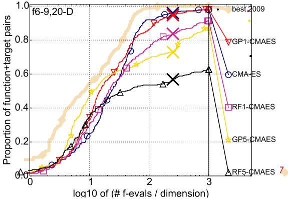


ECDF results on the whole BBOB (20 D)

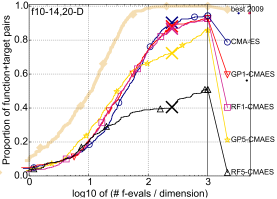
separable



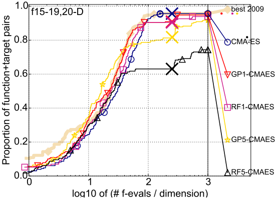
moderate



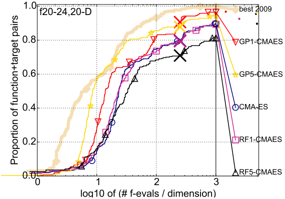
ill-conditional



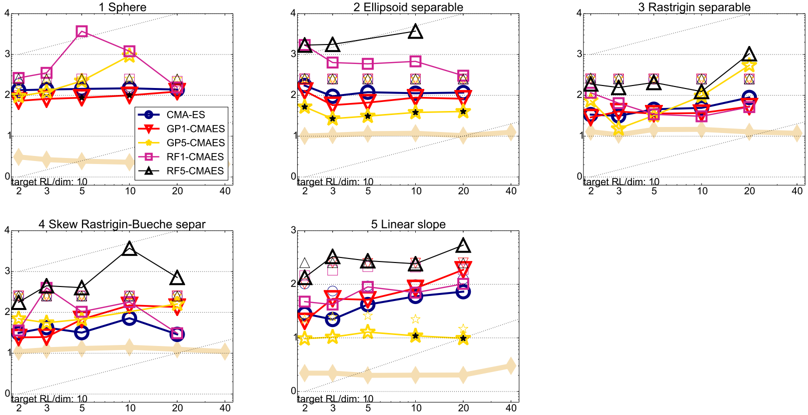
multi-modal



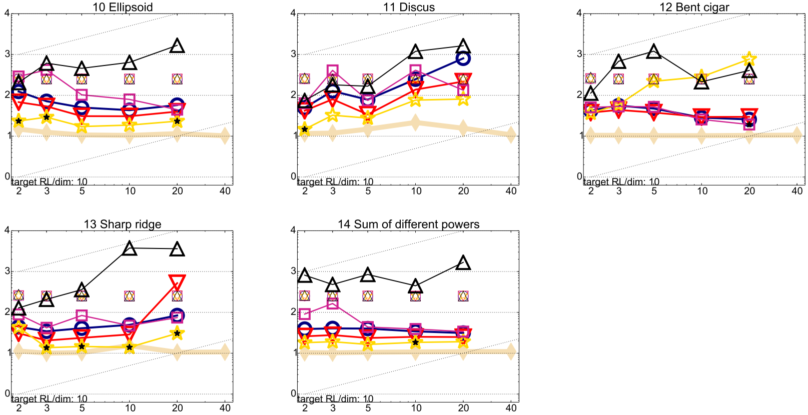
weakly structured multi-modal ill-conditional



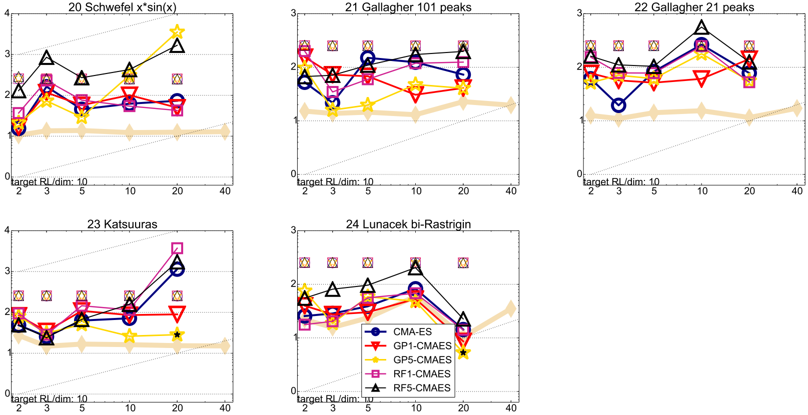
Results on separable BBOB functions (1–5)



Results on ill conditional BBOB functions (10–14)



Results on weakly structured multi-modal fcts (20–24)



Conclusions

- S-CMA-ES speeded-up CMA-ES on several BBOB functions
- **Gaussian processes** usually exhibit better performance than random forests
- **Random forests**' performance is rather balanced in 20D where Gaussian processes looses because of the high dimensionality
- Further investigation:
  - number of model generations adaptivity
  - reduction of the model training phase by starting from old parameters
  - random forest model precision

Thank you!

bajer at cs dot cas dot cz

pitra dot z at gmail dot com