Comparison of Ordinal and Metric Gaussian Process Regression as Surrogate Models for CMA Evolution Strategy

Anonymous Author(s)

ABSTRACT

In this paper, Gaussian processes are studied in connection with the state-of-the-art method for continuous black-box optimization CMA-ES. To combine them with the CMA-ES is challenging because CMA-ES invariance with respect to order preserving transformations suggests ordinal regression, whereas Gaussian process continuity suggests metric regression. Results of testing ordinal and metric Gaussian process regression, the former in 14 different settings, combined with the CMA-ES on noiseless benchmarks of the COCO platform are reported.

CCS CONCEPTS

•**Computing methodologies → Continuous space search;** *Model development and analysis; Uncertainty quantification;*

KEYWORDS

black-box optimization, evolutionary optimization, surrogate modelling, Gaussian-process regression

ACM Reference format:
Anonymous Author(s). 2017. Comparison of Ordinal and Metric Gaussian Process Regression
as Surrogate Models for CMA Evolution Strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17),* 8 pages.
DOI: 10.475/123_4

1 INTRODUCTION

An area of evolutionary optimization that is important from the point of view of real-world applications is *continuous black-box optimization*, i. e., optimization of functions for which no mathematical expression is known. The values of such black-box functions can be obtained only empirically, e. g., through measurements, experiments, or simulations. Such an empirical evaluation is sometimes very time-consuming or expensive: for example, in the applications of evolutionary algorithms to the optimization of chemical materials reported in [9], the evaluation of one generation takes several days to weeks and costs several to many thousands of euros. Needless to say, such situations are undesirable in the context of evolutionary optimization, where many fitness evaluations are typically needed. Therefore, *surrogate regression models* replacing the original expensive fitness in a part (typically, a large majority) of the evaluated points have been in use since the early 2000s (cf. the survey paper [10]). If we restrict attention to single-objective continuous black-box optimization, then basically the following four kinds of regression models have been employed to this end:

- **(i)** low degree polynomials [12]), which are models in the spirit of traditional *response surface models* [14];
- **(ii)** *artificial neural networks*, in particular multilayer perceptrons and radial basis function networks [1, 11];
- **(iii)** *support vector regression*, either metric or ordinal [13];
- **(iv)** *Gaussian processes (GP)*, a.k.a. kriging [15].

Whereas models of the first three kinds provide only estimates of the expected value of the original fitness, GP have the advantage of estimating the whole probability distribution of its values.

In this paper, we use Gaussian processes as surrogate models for the CMA-ES [5, 8]. To combine GP with the CMA-ES is challenging due to the fact that the CMA-ES is invariant with respect to order preserving transformations, suggesting ordinal regression, whereas the continuity of GP suggests metric regression. Therefore, the main objective of this paper is a comparison of ordinal and metric GP regression models as surrogate models for the CMA-ES. Because for ordinal GP regression, this is up to our knowledge the first time it is used for surrogate modelling, we investigate also the suitability of several different settings of the employed ordinal regression method to this end.

In the next section, the theoretical principles of the employed GP regression methods are recalled. Section3 sketches some details of our implementation of metric GP regression, and in Section 4, which is the core part of the paper, reports results of testing both kinds of GP regression in connection with the CMA-ES on the noiseless part of the COCO platform. Finally, the paper closes with conclusions drawn from the obtained results.

2 SURROGATE MODELS BASED ON METRIC AND ORDINAL GP-REGRESSION

A *Gaussian process* is a collection of random variables assigned to points of some Euclidean space (in our case a collection $(f(\mathbf{x}))_{\mathbf{x}\in\mathbb{R}^D}$, where $\mathbb{R}^D$ is the space containing the domain of the original fitness) such that each finite subcollection $(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n))$ has an n-dimensional normal, i. e., Gaussian distribution. Those distributions are parametrized by the mean and covariance, where the means are defined by the *mean function* $m : \mathbb{R}^D \to \mathbb{R}$ (often chosen as a constant), and the covariance matrix is a superposition of $(\mathbf{K}(\mathbf{x}_i,\mathbf{x}_j))_{i,j=1}^n$ and of $\sigma_n^2\mathbf{I}$, where $\mathbf{K} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a given function called *covariance function* and σ_n^2 is the variance of an additive independent identically distributed noise.

Among the many covariance functions that have been proposed for Gaussian processes (cf. [16]) , we have considered only the two most commonly encountered:

GECCO '17, Berlin, Germany

DOI: 10.475/123_4

- The *squared exponential* covariance function, defined

$$\mathbf{K}_{\mathrm{SE}}(\mathbf{x},\mathbf{y}) = \sigma_f^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{y})^2}{2\ell^2}\right). \tag{1}$$

  $\mathbf{K}_{\mathrm{SE}}$ is parametrized by the vector of parameters $\boldsymbol{\theta} = (\sigma_f^2, \ell)$, where σ_f^2 is the variance of the noise-free signal, and the parameter ℓ is called *characteristic length-scale*. Because the covariance function is itself a parameter of the GP, its parameters $\boldsymbol{\theta}$ are called *hyperparameters* of the GP.
- The *Matérn covariance function,* more precisely Matérn covariance function with $\nu = \frac{5}{2}$, defined

$$\mathbf{K}_{\mathrm{Mat}}(\mathbf{x},\mathbf{y}) = \\ = \sigma_f^2\left(1 + \frac{\sqrt{5}|\mathbf{x}-\mathbf{y}|}{\ell} + \frac{5(\mathbf{x}-\mathbf{y})^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}|\mathbf{x}-\mathbf{y}|}{\ell}\right). \tag{2}$$

  It has the same vector of hyperparameters $\boldsymbol{\theta}$ as $\mathbf{K}_{\mathrm{SE}}$.

The maximal likelihood estimates of $\boldsymbol{\theta}$ are taken as the values of the hyperparameters.

Since the normal distribution is continuous, GP can be directly used for metric regression. Indeed, the conditional distribution of the random variable $f^* = f(\mathbf{x}^*)$ assigned to a new $\mathbf{x}^* \notin \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ on condition $(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n)) = \mathbf{y}$ is $\mathcal{N}(\bar{f^*}, \mathrm{Var}(f^*))$, where:

$$\bar{f^*} = m(\mathbf{x}^*) + \mathbf{K}^*_{(1,\ldots,n)}((\mathbf{K}_{i,j})_{i,j=1}^n + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}^\top, \tag{3}$$

$$\mathrm{Var}(f^*) = \mathbf{K}(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{K}^*_{(1,\ldots,n)}((\mathbf{K}_{i,j})_{i,j=1}^n + \sigma_n^2\mathbf{I})^{-1})\mathbf{K}^{*\top}_{(1,\ldots,n)}, \tag{4}$$

using the notation $\mathbf{K}^*_{(1,\ldots,n)} = (\mathbf{K}(\mathbf{x}^*,\mathbf{x}_1),\ldots,\mathbf{K}(\mathbf{x}^*,\mathbf{x}_n))$.

The situation with ordinal regression is much more difficult. If the response variable should be ordinal, but still governed by a GP, then it has to be derived through some discretizing transformation from a latent GP behind it. Up to our knowledge, there exist three approaches addressing that task: expectation propagation, maximum aposteriori probability, and partial least squares combined with leave-one-out cross-validation [2, 17, 18]. In [18], they have been compared on 9 data sets from the UCI Machine learning repository [19] and their average predictive performance was similar. Therefore, we decided to implement the probabilistic least squares ordinal regression (PLSOR) [18] because differently to the other two approaches, it does not resort to approximation, and also due to its comparatively easy Matlab implementability and integrability with existing Matlab GP implementations.

The approach consists in defining ordered non-overlapping intervals $I_1 = (-\infty, b_1], I_2 = (b_1, b_2], \ldots, I_r = (b_{r-1}, \infty)$ separated by the thresholds $-\infty < b_1 < \cdots < b_{r-1} < \infty$ in such a way that the values of the latent GP can be linearly mapped to them, and that to each interval, at least one of the values $y_1,\ldots,y_n$ from the training data is mapped. Describing that linear mapping of a random variable $f(\mathbf{x})$ as $\alpha_0 - \alpha f(\mathbf{x})$ and introducing the auxiliary threshold values $b_0 = -\infty, b_r = \infty$, the probability that a random variable $f(\mathbf{x})$ with probability distribution $\mathcal{N}(\mu,\sigma)$ is mapped to a particular interval $I_k, k = 1,\ldots,r$, is

$$P(f(\mathbf{x}) \in I_k) = \Phi\left(\frac{b_k - (\alpha_0 - \alpha\mu)}{\sqrt{1+\alpha^2\sigma^2}}\right) - \Phi\left(\frac{b_{k-1} - (\alpha_0 - \alpha\mu)}{\sqrt{1+\alpha^2\sigma^2}}\right) = \\ = \Phi\left(\frac{\alpha\mu + \beta_k}{\sqrt{1+\alpha^2\sigma^2}}\right) - \Phi\left(\frac{\alpha\mu + \beta_{k-1}}{\sqrt{1+\alpha^2\sigma^2}}\right), \tag{5}$$

where Φ is the distribution function of the standard normal distribution $\mathcal{N}(0,1)$ and $\beta_k = b_k - \alpha_0, k = 0,\ldots,r$. Notice that this probability depends only on the relative positions $\beta_k = b_k - \alpha_0$ of the thresholds with respect to the intercept α_0 of the linear mapping, not on the absolute positions of the thresholds, nor on the value of the intercept. Taking into account (5), the PLSOR approach estimates the likelihood of a particular $y_i = f(\mathbf{x}_i), i = 1,\ldots,n$ as the probability that the prediction of $f(\mathbf{x}_i)$ based on the remaining training data without $(\mathbf{x}_i, y_i)$ is mapped to the same interval $I_{y_i} = (\beta_{y_i - 1} + \alpha_0, \beta_{y_i} + \alpha_0)$ to which y_i is mapped. Denoting the mean of that prediction μ_{-i} and its variance σ^2_{-i}, computed like the mean and variance in (3) and (4) but with hyperparameters of the GP estimated only from the remaining training data, this leads to the final estimated likelihood of the observed assignment of the training data to the intervals $I_1,\ldots,I_r$:

$$\hat{\mathcal{L}}(y_i \in I_{y_i}, i = 1,\ldots,n | \mathbf{x}, \boldsymbol{\theta}, \alpha, \beta_1,\ldots,\beta_{r-1}) = \\ \prod_{i=1}^n \left(\Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i}}{\sqrt{1+\alpha^2\sigma^2_{-i}}}\right) - \Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i - 1}}{\sqrt{1+\alpha^2\sigma^2_{-i}}}\right)\right). \tag{6}$$

From (6), both the hyperparameters $\boldsymbol{\theta}$ and the relative positions $\beta_1,\ldots,\beta_{r-1}$ of thresholds are simultaneously estimated.

Algorithm 1 Ordinal GP model training

Input: $(\mathbf{x}_i, y_i)_{i=1}^n$ (training points),
r (the number of bins for clustering),
$\boldsymbol{\theta}$ (initial values of latent GP hyperparameters),
α, $\{\beta_j\}_{j=1}^{r-1}$ (initial values of PLSOR hyperparameters)

1: $\{y_i^{\mathrm{ord}}\}_{i=1}^n \leftarrow \mathrm{cluster}(\{y_i\}_{i=1}^n, r)$

2: $(\alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})^* \leftarrow \underset{\alpha,\{\beta_j\}_{j=1}^{r-1},\boldsymbol{\theta}}{\arg\max} \log \hat{\mathcal{L}}(\{y_i^{\mathrm{ord}}\}_{i=1}^n | \mathbf{x}, \alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})$ (see Eq. (6))

Output: $(\alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})^*$ (trained model hyperparameters)

3 IMPLEMENTATION OF ORDINAL GP

In this section, we present implementation details of ordinal GP model for the DTS-CMA-ES [15].

The ordinal GP model-building phase, depicted in Algorithm 1, starts with clustering of the input data $(\mathbf{x}_i, y_i)_{i=1}^n$ to intervals $I_1,\ldots,I_r$. After that, the hyperparameters are optimized by likelihood (6) maximization via leave-one-out cross-validation. The values maximizing (6) determine the trained ordinal GP model.

Two kinds of clustering according to the fitness values are implemented: *quantile clustering* and log-scaled *agglomerative hierarchical clustering*.

The ordinal GP model prediction procedure is depicted in Algorithm 2. The prediction of the ordinal class q_i of a point $\mathbf{x}_i$ is calculated as the weighted average of ordinal class values weighted by the classes' probabilities p_i^k. The output of the GP model is

Algorithm 2 Ordinal GP model prediction

Input: $\{\mathbf{x}_i\}_{i=1}^{\lambda}$ (population of points),
$\boldsymbol{\theta}$ (trained latent GP hyperparameters),
α, $\{\beta_j\}_{j=1}^{r-1}$ (trained PLSOR hyperparameters)

$p_i^k \leftarrow P(f(\mathbf{x}_i) \in I_k | \mathbf{x}_i, \alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})$ $\quad \forall i = 1, \ldots, \lambda$ $\quad \forall k = 1, \ldots, r$ (see Eq. (5))

$q_i \leftarrow \sum_{k=1}^{r} p_i^k k$ $\quad \forall i = 1, \ldots, \lambda$

$\{\mathbf{x}_{i:\lambda}\}_{i=1}^{\lambda} \leftarrow \text{order}(\{\mathbf{x}_i\}_{i=1}^{\lambda})$, where $q_{1:\lambda} \le q_{2:\lambda} \le \cdots \le q_{\lambda:\lambda}$

Output: $\{\mathbf{x}_{i:\lambda}\}_{i=1}^{\lambda}$ (ordered population)

the ordered set of CMA-ES generated population $\{\mathbf{x}_{i:\lambda}\}_{i=1}^{\lambda}$, where the index $i{:}\lambda$ denotes the index of the i-th ranked point, that is $q_{1:\lambda} \le q_{2:\lambda} \le \cdots \le q_{\lambda:\lambda}$.

4 EXPERIMENTS ON THE COCO PLATFORM

First, our PLSOR implementation is validated on UCI datasets [2] to verify agreement with the original implementation of PLSOR [18], which was not available. Second, both models are compared without combining with the CMA-ES on datasets collected from the DTS-CMA-ES [15] runs on the noiseless part of the COCO/BBOB framework [6, 7] to compare their predictive accuracy. Finally, we compare the performances of three different ordinal model settings to DTS-CMA-ES, and the original CMA-ES [5] on the BBOB benchmark functions.

4.1 Validation of our PLSOR implementation

The PLSOR method was benchmarked on a collection of 9 datasets from the UCI machine learning repository similarly to previous approaches to ordinal regression with Gaussian processes [2, 18].

A 20-fold cross-validation was used and the response variables were discretized into either 5 or 10 ordinal categories by equal frequency binning. The PLSOR performance was measured with the *zero-one error* (ZOE), i. e., the ratio of incorrect test predictions to the number of the test data, $\frac{1}{t}\sum_{i=1}^{t}\mathbb{I}(y_i \neq \hat{y}_i)$, where $\mathbb{I}(\cdot)$ is the indicator function.

Table 1 compares ZOE means and standard deviations on the 5-categories versions of the benchmark datasets with the results reported in [18].

Data	ZOE (original)	ZOE (implementation)
Diabetes	**0.48** ± 0.11	0.57 ± 0.11
Pyrimidine	0.39 ± 0.09	**0.36** ± 0.07
Triazines	0.54 ± 0.03	**0.54** ± 0.03
Wisconsin	**0.66** ± 0.03	0.68 ± 0.05
Machine	**0.18** ± 0.03	0.19 ± 0.04
AutoMPG	**0.26** ± 0.02	0.26 ± 0.02
Boston	**0.25** ± 0.03	0.25 ± 0.03
Stocks	0.11 ± 0.02	**0.11** ± 0.02
Abalone	0.22 ± 0.03	**0.22** ± 0.04

Table 1: Validation of the PLSOR implementation on the 5 categories versions of the benchmark datasets. Mean and standard deviations of the zero-one error over 20 cross-validation sets reproduced from [18] (middle column) and values for our implementation (right column). The lower mean values are highlighted in bold.

We observe that our implementation is clearly worse only on two datasets, Diabetes and Wisconsin. On the remaining datasets, it slightly exceeds or comes very close to the referential results. According to the Wilcoxon signed-ranks test, the difference between both implmentations is not significant (p=value = 0.16).

4.2 Predictive accuracy of metric and ordinal GP regression

As our primary interest is in using Gaussian processes as surrogate models for the CMA-ES, we tested the proposed models on datasets corresponding to the 24 noiseless COCO benchmarks while comparing the models' capabilities to predict the ordering of a new population.

The training datasets were collected for each function ($f \in 1, 2, \ldots, 24$) and dimension ($D \in 2, 5, 10$) from 10 snapshots of the DTS-CMA-ES *archive* (i. e., the set of so-far originally-evaluated points). These snapshots were taken equidistantly throughout CMA-ES generations and each testing dataset was sampled using a simple combination of the CMA-ES state variables (m, σ, C), which assures equal distribution of its training and test part.

We have compared DTS-CMA-ES' metric GP models with 14 settings of the PLSOR. These settings differ with respect to:

(i) covariance function – $\mathbf{K}_{\mathrm{SE}}$ (used in [18]) and $\mathbf{K}_{\mathrm{Mat}}$;

(ii) type of obtaining ordinal from continuous f-value – quantile clustering, agglomerative hierarchical clustering, or direct ordering of f-values (no clustering);

(iii) the number of clusters in (ii) – μ, λ or 2λ.

Due to space limitations, only results for $D = 5$ are presented, in Table 3. As can be seen, the PLSOR models in general produce higher ZOE than metric GP. An exception is the datasets for the function f_6 (Attractive sector) where standard continuous regression models fail to regress a sharp edge where the true optimum of the function is located and ordinal models seem to benefit from their invariance on the smoothness of the corresponding function.

4.3 Metric and ordinal GP surrogate models for the CMA-ES

Based on the off-line model testing, three well-performing ordinal GP models were chosen for the COCO/BBOB noiseless benchmarking in connection with the DTS-CMA-ES algorithm: Ord-N-DTS, Ord-Q-DTS, and Ord-H-DTS, where N denotes no clustering, Q quantile-based clustering, and H hierarchical agglomerative clustering. The number of ordinal classes for clustering was set the same as the population size λ. The ordinal DTS-CMA-ES versions were tested using the function values as a criterion for choosing the points for original function re-evaluation. The remainder of the parameters were left the same as in the original DTS-CMA-ES settings.

The original DTS-CMA-ES was employed using the GP prediction variance as the uncertainty criterion, the population size $\lambda = 8 + \lfloor 6 \log D \rfloor$, and the number of originally-evaluated points $n_{\mathrm{orig}} = \lceil 0.05\lambda \rceil$. The remaining parameters were taken from [15].

The original CMA-ES was tested in its IPOP-CMA-ES version (Matlab code v. 3.61) with the following settings: the number of restarts = 4, IncPopSize = 2, $\sigma_{\text{start}} = \frac{8}{3}$, $\lambda = 4 + \lfloor 3 \log D \rfloor$. The remaining settings were left default.

The results in Figures 1 and 2 and in Table 4 show the effect of usage of the PLSOR models instead of the metric GP in the DTS-CMA-ES optimizer on all the 24 noiseless COCO/BBOB benchmarks [6, 7]. The **expected running time (ERT)**, used in the figures and tables, depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, and is computed over all relevant trials as the number of the original function evaluations (FEs) executed during each trial until the best function value reached f_t, summed over all trials and divided by the number of trials that actually reached f_t [6]. More detailed results can be found on an authors' webpage[1].

The effect of different clustering methods was not shown important (see Fig. 2). The ordinal DTS-CMA-ES outperforms the metric version only on several (mostly multi-modal) functions in lower dimensions (on f_6, f_{16-19}, and f_{21-22}). While, the original DTS-CMA-ES is dominant on the rest of tested functions and higher dimensions.

The considerably lower speedup of PLSOR-based DTS-CMA-ES compared to the original DTS-CMA-ES follows the results from the previous subsection.

We test statistical significance of performance differences in $5D$ using Iman and Davenport's improvement of the Friedman test [3]. The test is conducted separately for two function evaluation budgets.

Let us denote by Δ_f^{med} the empirical median of distance to optimum Δ_f over all function instances depending on number of function evaluations. Further, let $\#\text{FE}_\text{T}$ be the smallest number of function evaluations at which at least one algorithm reached the precision $\Delta_f^{\text{med}} \leq 10^{-8}$, or $\#\text{FE}_\text{T} = 100D$ if no algorithm reached the precision within $100D$ evaluations. The algorithms are ranked on each BBOB function with respect to Δ_f^{med} at a given budget of function evaluations. The null hypothesis of equal performance of all algorithms is rejected at a higher function evaluation budget $\#\text{FEs} = \#\text{FE}_\text{T}$ ($p < 10^{-3}$), as well as at a lower budget $\#\text{FEs} = \frac{\#\text{FE}_\text{T}}{3}$ ($p < 10^{-3}$).

We test pairwise differences in algorithms' performance using the post-hoc Friedman test [4] with two different procedures for controlling the family-wise error. The Bonferroni procedure simply divides the significance level by the number of tested hypotheses, whereas the less conservative Bergmann-Hommel procedure corrects the significance level per each logically consistent family of hypotheses. The results of both multiple comparisons are reported in Table 2. There is no significant effect of clustering on ordinal regression DTS performance in $5D$ at significance level $\alpha = 0.05$. On the other hand, the metric regression DTS significantly outperforms the ordinal regression DTS at both tested budgets of function evaluations.

4.4 CPU Timing

In order to evaluate the CPU timing of the algorithms, we have run the Ord-Q-DTS on the BBOB test suite with restarts for a maximum budget equal to $100D$ function evaluations. The MATLAB code was run in a single thread on the Metacentrum grid with CPUs from the Intel Xeon family. The time per function evaluation on f_8 for dimensions 2, 3, 5, 10 equals 4.15, 6.48, 12.48 and 13.95 seconds respectively.

[1] http://uivty.cs.cas.cz/~cma/gecco2017bbob/

5 CONCLUSIONS

In this paper, we have compared the ordinal GP regression model using PLSOR implementation with the metric GP regression model used in DTS-CMA-ES. The comparison of our implementation of the PLSOR method reproduced the published results on the UCI datasets. On the other hand, the usage of the PLSOR models as surrogates for the CMA-ES was not shown as straightforward on the BBOB benchmark: the performance of the PLSOR models is considerably lower than the standard GP models with few exceptions, especially on the *attractive sector* function f_6.

A possible perspective is to improve ordinal GP models by implementing clustering methods using informations from the GP kernels. Another perspective is to further develop DTS-CMA-ES to adaptively switch from metric to ordinal GP regression model when the metric prediction is not succesful (e. g., on f_6).

REFERENCES

[1] L. Bajer and M. Holeňa. 2010. Surrogate Model for Continuous and Discrete Genetic Optimization Based on RBF Networks. In *Intelligent Data Engineering and Automated Learning. Lecture Notes in Computer Science 6283*. Springer, 251–258.
[2] W. Chu and Z. Ghahramani. 2005. Gaussian Processes for Ordinal Regression. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1019–1041.
[3] J. Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7 (Dec. 2006), 1–30.
[4] S. García and F. Herrera. 2008. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *J. Mach. Learn. Res.* 9 (2008), 2677–2694.
[5] N. Hansen. 2006. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation*. Springer, 75–102.
[6] N. Hansen, A. Auger, S. Finck, and R. Ros. 2012. *Real-Parameter Black-Box Optimization Benchmarking 2012: Experimental Setup.* Technical Report. INRIA.
[7] N. Hansen, S. Finck, R. Ros, and A. Auger. 2009. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions.* Technical Report RR-6829. INRIA. Updated February 2010.
[8] N. Hansen and A. Ostermaier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9 (2001), 159–195.
[9] M. Holeňa, D. Linke, and L. Bajer. 2012. Surrogate Modeling in the Evolutionary Optimization of Catalytic Materials. In *GECCO 2012*. 1095–1102.
[10] Y. Jin. 2011. Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges. *Swarm and Evolutionary Computation* 1 (2011), 61–70.
[11] Y. Jin, M. Hüsken, M. Olhofer, and Sendhoff B. 2005. Neural Networks for Fitness Approximation in Evolutionary Optimization. In *Knowledge Incorporation in Evolutionary Computation*. Springer, 281–306.
[12] S. Kern, N. Hansen, and P. Koumoutsakos. 2006. Local Meta-models for Optimization Using Evolution Strategies. In *Parallel Problem Solving from Nature - PPSN IX (LNCS)*, Vol. 4193. Springer Berlin Heidelberg, 939–948.
[13] I. Loshchilov, M. Schoenauer, and M. Sebag. 2013. Intensive Surrogate Model Exploitation in Self-Adaptive Surrogate-Assisted CMA-ES (saACM-ES). In *GECCO 2013*. 439–446.
[14] R.H. Myers, D.C. Montgomery, and C.M. Anderson-Cook. 2009. *Response Surface Methodology: Proces and Product Optimization Using Designed Experiments.* John Wiley and Sons, Hoboken.
[15] Z. Pitra, L. Bajer, and M. Holeňa. 2016. *Doubly Trained Evolution Control for the Surrogate CMA-ES.* Springer International Publishing, Cham, 59–68.
[16] E. Rasmussen and C. Williams. 2006. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge.
[17] P.K. Srijith, S. Shevade, and S. Sundararajan. 2012. Validation Based Sparse Gaussian Processes for Ordinal Regression. In *ICONIP 2012*. 409–416.
[18] P. K. Srijith, S. Shevade, and S. Sundararajan. 2012. *A Probabilistic Least Squares Approach to Ordinal Regression.* Springer Berlin Heidelberg, 683–694.
[19] University of California, Irvine 2016. UCI Repository of Machine Learning Databases. (2016). http://www.ics.uci.edu/~mlearn.

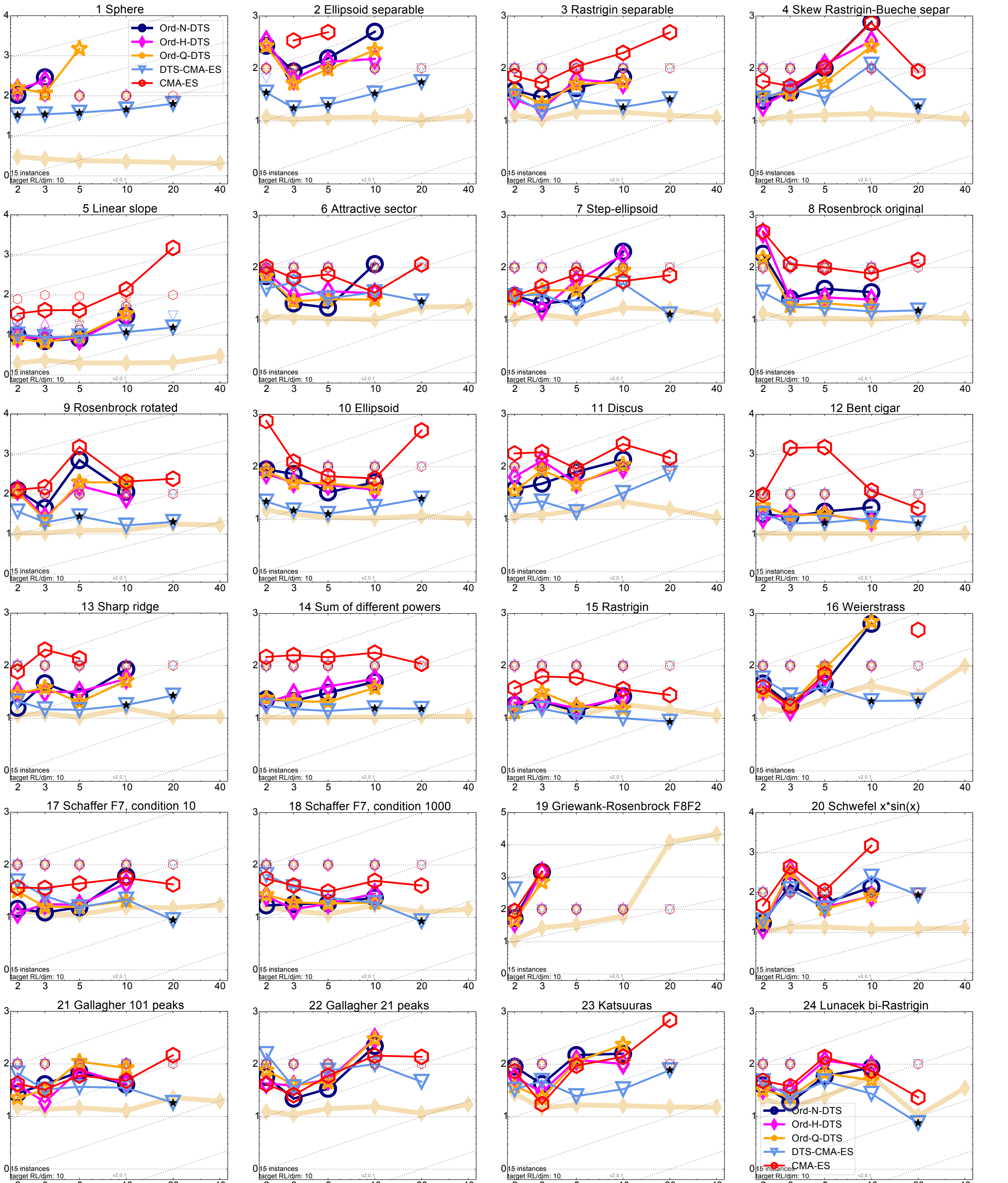

Figure 1: Average running time (aRT in number of f-evaluations as $\log_{10}$ value) divided by dimension versus dimension. The target function value is chosen such that the best algorithm from BBOB 2009 just failed to achieve an aRT of $10 \times \mathrm{DIM}$. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○: Ord-N-DTS, ◇: Ord-H-DTS, ⋆: Ord-Q-DTS, ▽: DTS-CMA-ES, ⬡: CMA-ES

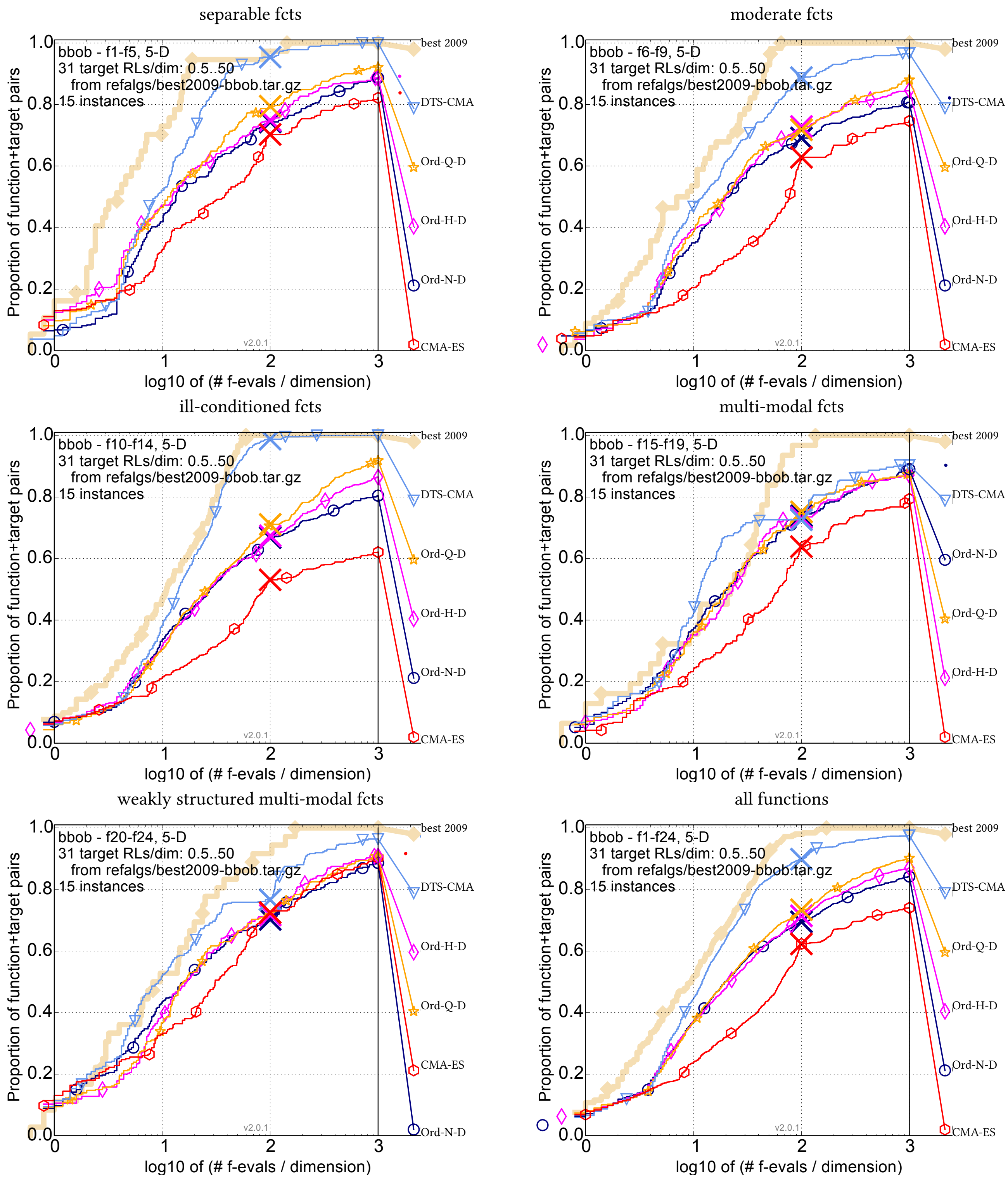

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 5-D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times$ DIM, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 50\}$. The "best 2009" line corresponds to the best aRT observed during BBOB 2009 for each selected target.

5*D*	Ord-N-DTS		Ord-H-DTS		Ord-Q-DTS		DTS-CMA-ES		CMA-ES	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
Ord-N-DTS	—	—	11	9	10	6	6	1	23^{*}	7
Ord-H-DTS	12	15	—	—	9	9	4	3	22^{*}	10
Ord-Q-DTS	13	18	14	15	—	—	3	3	22^{*}	10
DTS-CMA-ES	18^{**}	23^{*}	20^{*}	21^{*}	21^{**}	21^{*}	—	—	22^{*}	17^{**}
CMA-ES	1	17^{**}	2	14	2	14	2	7	—	—

Table 2: A pairwise comparison of the algorithms in $5D$ **over the BBOB for different evaluation budgets. The number of wins of** i**-th algorithm against** j**-th algorithm over all benchmark functions is given in** i**-th row and** j**-th column. The asterisk marks the row algorithm being significantly better than the column algorithm according to the Friedman post-hoc test with the Bonferroni correction at family-wise significance level** $\alpha = 0.05$**. The double asterisk marks significant results at the same significance level according to the Friedman test with more powerful Bergmann-Hommel correction of family-wise error. The Bergmann-Hommel procedure rejects more hypotheses, as it exploits logical relations between them.**

5D	DTS – GP	K_{Mat}	K_{Mat},H,μ	K_{Mat},H,λ	K_{Mat},H,2λ	K_{Mat},Q,μ	K_{Mat},Q,λ	K_{Mat},Q,2λ	K_{SE}	K_{SE},H,μ	K_{SE},H,λ	K_{SE},H,2λ	K_{SE},Q,μ	K_{SE},Q,λ	K_{SE},Q,2λ
f_1	0.84 ± 0.20	0.91 ± 0.12	0.86 ± 0.27	0.93 ± 0.08	0.89 ± 0.12	0.85 ± 0.25	0.82 ± 0.20	0.89 ± 0.12	0.79 ± 0.21	0.84 ± 0.21	0.79 ± 0.22	0.87 ± 0.21	0.88 ± 0.11	**0.76 ± 0.18**	0.82 ± 0.21
f_2	**0.73 ± 0.18**	0.89 ± 0.10	0.90 ± 0.09	0.90 ± 0.09	0.90 ± 0.09	0.81 ± 0.16	0.87 ± 0.14	0.93 ± 0.08	0.91 ± 0.07	0.91 ± 0.11	0.83 ± 0.12	0.91 ± 0.09	0.86 ± 0.12	0.93 ± 0.08	0.88 ± 0.11
f_3	0.88 ± 0.11	0.85 ± 0.11	0.87 ± 0.15	0.92 ± 0.12	0.89 ± 0.12	0.91 ± 0.08	**0.84 ± 0.10**	0.90 ± 0.08	0.86 ± 0.16	0.88 ± 0.13	0.89 ± 0.07	0.92 ± 0.08	0.88 ± 0.07	0.90 ± 0.05	0.92 ± 0.06
f_4	**0.75 ± 0.38**	0.92 ± 0.05	0.93 ± 0.08	0.92 ± 0.09	0.93 ± 0.07	0.90 ± 0.08	0.90 ± 0.07	0.92 ± 0.06	0.89 ± 0.11	0.90 ± 0.08	0.91 ± 0.08	0.86 ± 0.10	0.85 ± 0.11	0.90 ± 0.07	0.92 ± 0.07
f_5	**0.00 ± 0.00**	0.71 ± 0.19	0.72 ± 0.24	0.71 ± 0.22	0.72 ± 0.19	0.60 ± 0.21	0.67 ± 0.28	0.63 ± 0.25	0.66 ± 0.19	0.71 ± 0.27	0.69 ± 0.24	0.65 ± 0.21	0.73 ± 0.22	0.63 ± 0.17	0.70 ± 0.23
f_6	0.89 ± 0.09	**0.55 ± 0.32**	0.83 ± 0.17	0.76 ± 0.21	0.60 ± 0.36	0.70 ± 0.30	0.66 ± 0.33	0.68 ± 0.31	0.65 ± 0.32	0.75 ± 0.29	0.76 ± 0.22	0.61 ± 0.29	0.84 ± 0.17	0.72 ± 0.31	0.78 ± 0.33
f_7	0.88 ± 0.12	0.84 ± 0.08	0.87 ± 0.08	0.81 ± 0.15	0.88 ± 0.08	0.85 ± 0.10	0.82 ± 0.09	**0.81 ± 0.06**	0.85 ± 0.08	0.85 ± 0.11	0.86 ± 0.09	0.85 ± 0.09	0.92 ± 0.07	0.93 ± 0.06	0.89 ± 0.10
f_8	**0.74 ± 0.23**	0.82 ± 0.13	0.89 ± 0.13	0.83 ± 0.12	0.87 ± 0.13	0.88 ± 0.09	0.86 ± 0.14	0.85 ± 0.13	0.83 ± 0.12	0.81 ± 0.14	0.86 ± 0.10	0.88 ± 0.18	0.90 ± 0.09	0.84 ± 0.14	0.76 ± 0.18
f_9	**0.70 ± 0.28**	0.78 ± 0.20	0.88 ± 0.12	0.85 ± 0.18	0.81 ± 0.24	0.79 ± 0.19	0.84 ± 0.19	0.82 ± 0.19	0.81 ± 0.17	0.86 ± 0.13	0.78 ± 0.12	0.83 ± 0.17	0.85 ± 0.15	0.81 ± 0.17	0.74 ± 0.18
f_{10}	0.81 ± 0.19	0.81 ± 0.18	0.88 ± 0.11	**0.78 ± 0.19**	0.81 ± 0.19	0.83 ± 0.12	0.82 ± 0.19	0.83 ± 0.18	0.86 ± 0.11	0.85 ± 0.13	0.84 ± 0.14	0.82 ± 0.14	0.93 ± 0.07	0.87 ± 0.20	0.82 ± 0.18
f_{11}	**0.81 ± 0.12**	0.88 ± 0.15	0.93 ± 0.08	0.91 ± 0.07	0.95 ± 0.07	0.91 ± 0.08	0.90 ± 0.07	0.88 ± 0.13	0.91 ± 0.08	0.91 ± 0.08	0.93 ± 0.07	0.93 ± 0.08	0.93 ± 0.08	0.90 ± 0.09	0.92 ± 0.07
f_{12}	**0.67 ± 0.32**	0.86 ± 0.14	0.86 ± 0.12	0.90 ± 0.12	0.87 ± 0.13	0.92 ± 0.10	0.91 ± 0.10	0.86 ± 0.12	0.87 ± 0.14	0.85 ± 0.14	0.91 ± 0.11	0.87 ± 0.13	0.88 ± 0.11	0.81 ± 0.14	0.85 ± 0.10
f_{13}	**0.69 ± 0.28**	0.87 ± 0.13	0.85 ± 0.12	0.89 ± 0.10	0.83 ± 0.12	0.82 ± 0.15	0.80 ± 0.17	0.85 ± 0.13	0.84 ± 0.15	0.89 ± 0.10	0.91 ± 0.09	0.79 ± 0.11	0.84 ± 0.17	0.80 ± 0.11	0.82 ± 0.10
f_{14}	0.83 ± 0.13	0.83 ± 0.17	0.90 ± 0.13	0.84 ± 0.15	0.85 ± 0.17	0.84 ± 0.12	0.92 ± 0.12	0.92 ± 0.13	0.88 ± 0.16	0.88 ± 0.09	0.83 ± 0.16	0.86 ± 0.12	0.86 ± 0.14	**0.82 ± 0.12**	0.90 ± 0.12
f_{15}	0.78 ± 0.25	0.91 ± 0.07	0.94 ± 0.06	0.90 ± 0.10	0.92 ± 0.11	0.90 ± 0.10	0.82 ± 0.21	0.89 ± 0.11	0.93 ± 0.07	**0.78 ± 0.31**	0.90 ± 0.09	0.92 ± 0.08	0.91 ± 0.08	0.89 ± 0.09	0.95 ± 0.05
f_{16}	0.92 ± 0.15	0.89 ± 0.11	0.91 ± 0.07	**0.84 ± 0.16**	0.93 ± 0.08	0.87 ± 0.11	0.90 ± 0.16	0.89 ± 0.16	0.91 ± 0.06	0.93 ± 0.10	0.96 ± 0.04	0.95 ± 0.06	0.88 ± 0.18	0.86 ± 0.15	0.89 ± 0.13
f_{17}	0.87 ± 0.15	0.85 ± 0.13	0.88 ± 0.11	0.89 ± 0.12	0.85 ± 0.13	0.89 ± 0.08	0.87 ± 0.14	0.85 ± 0.13	0.91 ± 0.09	0.91 ± 0.11	0.89 ± 0.08	0.94 ± 0.06	0.93 ± 0.08	**0.85 ± 0.12**	0.93 ± 0.07
f_{18}	**0.82 ± 0.18**	0.90 ± 0.07	0.90 ± 0.09	0.87 ± 0.10	0.89 ± 0.10	0.85 ± 0.09	0.88 ± 0.08	0.89 ± 0.10	0.88 ± 0.11	0.92 ± 0.10	0.95 ± 0.06	0.94 ± 0.10	0.93 ± 0.05	0.89 ± 0.09	0.88 ± 0.13
f_{19}	0.72 ± 0.34	**0.62 ± 0.26**	0.76 ± 0.21	0.63 ± 0.18	0.71 ± 0.23	0.88 ± 0.09	0.69 ± 0.24	0.68 ± 0.25	0.71 ± 0.27	0.78 ± 0.25	0.74 ± 0.18	0.78 ± 0.22	0.87 ± 0.08	0.74 ± 0.26	0.65 ± 0.33
f_{20}	**0.63 ± 0.33**	0.86 ± 0.13	0.82 ± 0.15	0.85 ± 0.17	0.87 ± 0.13	0.90 ± 0.10	0.85 ± 0.17	0.87 ± 0.18	0.88 ± 0.16	0.86 ± 0.14	0.83 ± 0.16	0.81 ± 0.15	0.90 ± 0.05	0.82 ± 0.16	0.86 ± 0.16
f_{21}	**0.70 ± 0.38**	0.90 ± 0.12	0.93 ± 0.07	0.83 ± 0.17	0.88 ± 0.11	0.87 ± 0.15	0.87 ± 0.13	0.91 ± 0.11	0.89 ± 0.10	0.88 ± 0.09	0.87 ± 0.11	0.87 ± 0.09	0.94 ± 0.14	0.86 ± 0.08	0.84 ± 0.13
f_{22}	**0.70 ± 0.36**	0.76 ± 0.23	0.87 ± 0.08	0.88 ± 0.08	0.82 ± 0.22	0.80 ± 0.25	0.80 ± 0.18	0.75 ± 0.25	0.89 ± 0.08	0.86 ± 0.13	0.85 ± 0.17	0.91 ± 0.13	0.84 ± 0.22	0.83 ± 0.15	0.86 ± 0.13
f_{23}	0.75 ± 0.33	0.72 ± 0.39	0.78 ± 0.29	0.83 ± 0.23	**0.68 ± 0.38**	0.78 ± 0.30	0.68 ± 0.39	0.73 ± 0.31	0.79 ± 0.32	0.83 ± 0.17	0.86 ± 0.16	0.84 ± 0.17	0.84 ± 0.23	0.84 ± 0.18	0.83 ± 0.20
f_{24}	**0.77 ± 0.22**	0.88 ± 0.06	0.90 ± 0.11	0.88 ± 0.10	0.87 ± 0.09	0.90 ± 0.09	0.84 ± 0.11	0.91 ± 0.05	0.92 ± 0.09	0.91 ± 0.14	0.88 ± 0.13	0.95 ± 0.06	0.93 ± 0.05	0.91 ± 0.09	0.92 ± 0.07
	0.74 ± 0.29	0.82 ± 0.19	0.87 ± 0.15	0.85 ± 0.15	0.84 ± 0.19	0.84 ± 0.17	0.83 ± 0.19	0.84 ± 0.18	0.84 ± 0.17	0.86 ± 0.16	0.85 ± 0.15	0.85 ± 0.16	0.88 ± 0.13	0.84 ± 0.16	0.84 ± 0.17

Table 3: Means and standard deviations of ZOE of offline testing on 10 selected generations for 24 noiseless functions and $5D$, DTS – GP**: a metric GP regression model,** K_{SE} **and** K_{Mat}**: squared-exponential and Matérn covariance function,** Q**: quantile-based clustering,** H**: hierarchical agglomerative clustering,** μ, λ, 2λ**: the number of ordinal classes for clustering. The last row shows mean and standard deviation through all functions. The best achieved values in each function are given in bold.**

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*2.5e+1*:5.0	*1.6e+1*:8.0	*1.0e-8*:12	*1.0e-8*:12	*1.0e-8*:12	15/15
Ord-N	2.9(2)	2.5(0.4)	∞	∞	∞ *501*	0/15
Ord-H	2.9(2)	2.6(1)	∞	∞	∞ *501*	0/15
Ord-Q	**2.9**(2)	2.5(2)	627(460)	627(491)	627(356)	1/15
DTS-C	3.4(2)	**2.5**(0.8)	**16**(0.9)$^{\star 4}$	**16**(0.9)$^{\star 4}$	**16**(1)$^{\star 4}$	15/15
CMA-E	4.6(4)	4.5(6)	∞	∞	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f2	*1.6e+6*:3.0	*4.0e+5*:11	*2.5e+4*:16	*6.3e+2*:58	*1.0e-8*:95	15/15
Ord-N	3.3(4)	2.0(2)	5.7(3)	13(15)	∞ *501*	0/15
Ord-H	1.9(4)	**1.4**(2)	4.7(3)	12(27)	∞ *501*	0/15
Ord-Q	2.8(3)	1.7(1)	5.9(3)	8.3(10)	∞ *501*	0/15
DTS-C	**1.6**(2)	1.6(1)	**2.6**(1)	**1.8**(0.7)$^{\star 3}$	**4.6**(0.3)$^{\star 4}$	14/15
CMA-E	3.8(5)	2.5(2)	14(8)	42(67)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f3	*1.6e+2*:4.0	*1.0e+2*:15	*6.3e+1*:23	*2.5e+1*:73	*1.0e+1*:716	15/15
Ord-N	3.9(3)	1.6(0.8)	2.0(0.7)	2.9(3)	10(11)	1/15
Ord-H	**2.6**(2)	1.6(2)	2.4(1)	4.2(4)	2.4(2)	4/15
Ord-Q	3.3(3)	**1.5**(0.6)	1.7(0.6)	3.4(3)	3.4(4)	3/15
DTS-C	4.2(4)	1.9(0.9)	**1.7**(0.7)	**1.7**(0.4)	**0.66**(0.8)	9/15
CMA-E	3.5(2)	2.2(2)	4.0(4)	7.4(13)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f4	*2.5e+2*:3.0	*1.6e+2*:10	*1.0e+2*:19	*4.0e+1*:65	*1.6e+1*:434	15/15
Ord-N	4.6(4)	3.3(3)	4.8(7)	7.8(12)	∞ *501*	0/15
Ord-H	**2.6**(4)	**2.1**(2)	2.5(2)	9.3(6)	∞ *501*	0/15
Ord-Q	3.1(4)	2.5(4)	**2.4**(3)	4.1(4)	17(15)	1/15
DTS-C	4.2(4)	2.7(2)	2.5(2)	**2.1**(3)	**1.7**(0.9)	7/15
CMA-E	2.8(3)	3.5(4)	5.1(5)	7.9(4)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f5	*6.3e+1*:4.0	*4.0e+1*:10	*1.0e-8*:10	*1.0e-8*:10	*1.0e-8*:10	15/15
Ord-N	3.8(2)	2.0(0.5)	4.1(2)	4.1(2)	4.1(2)	15/15
Ord-H	**3.5**(2)	1.9(0.2)	**3.8**(2)	**3.8**(1)	**3.8**(2)	15/15
Ord-Q	**3.5**(2)	**1.6**(0.1)	4.6(2)	4.6(3)	4.6(4)	15/15
DTS-C	3.8(2)	2.0(0.4)	4.6(1)	4.6(2)	4.6(2)	15/15
CMA-E	4.0(6)	3.6(3)	21(18)	21(19)	21(11)	15/15

#FEs/D	0.5	1.2	3	10	50	#succ
f6	*1.0e+5*:3.0	*2.5e+4*:8.0	*1.0e+2*:16	*2.5e+1*:54	*2.5e-1*:254	15/15
Ord-N	**2.4**(4)	1.9(3)	2.1(2)	**1.6**(0.7)	∞ *502*	0/15
Ord-H	3.1(4)	1.8(1)	2.2(2)	3.3(4)	∞ *501*	0/15
Ord-Q	2.8(3)	**1.6**(2)	2.8(2)	2.3(0.9)	∞ *502*	0/15
DTS-C	2.5(3)	1.8(1)	**1.6**(0.8)	2.4(3)	∞ *500*	0/15
CMA-E	5.4(0.3)	4.6(4)	6.3(4)	6.9(2)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f7	*1.6e+2*:4.0	*6.3e+1*:11	*2.5e+1*:20	*4.0e+0*:54	*1.0e+0*:324	15/15
Ord-N	3.9(2)	2.5(3)	2.3(1)	2.2(1)	1.4(3)	10/15
Ord-H	3.1(3)	2.3(1)	2.5(2)	5.3(7)	5.2(5)	4/15
Ord-Q	4.2(3)	2.5(1)	2.1(2)	3.4(1)	2.4(3)	7/15
DTS-C	**2.3**(3)	**1.9**(0.7)	**1.4**(0.6)	**1.6**(1)	**1.0**(0.9)	13/15
CMA-E	4.3(7)	4.7(4)	6.8(7)	6.9(0.7)	2.4(0.0)	9/15

#FEs/D	0.5	1.2	3	10	50	#succ
f8	*1.0e+4*:5.0	*6.3e+3*:7.0	*1.0e+3*:18	*6.3e+1*:54	*1.6e+0*:258	15/15
Ord-N	3.7(3)	3.0(1)	2.0(2)	3.7(7)	∞ *501*	0/15
Ord-H	**2.4**(2)	**2.2**(1)	**1.5**(0.8)	2.5(2)	28(32)	1/15
Ord-Q	2.6(2)	2.2(2)	1.9(0.9)	1.9(0.6)	28(29)	1/15
DTS-C	2.8(2)	2.4(2)	1.9(0.4)	**1.6**(1)	**1.1**(0.6)$^{\star 2}$	13/15
CMA-E	4.0(3)	4.4(2)	6.6(6)	9.3(7)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f9	*2.5e+1*:20	*1.6e+1*:26	*1.0e+1*:35	*4.0e+0*:62	*1.6e-2*:256	15/15
Ord-N	13(8)	14(11)	17(8)	56(34)	∞ *501*	0/15
Ord-H	7.0(7)	7.0(8)	6.0(1)	13(14)	∞ *501*	0/15
Ord-Q	7.3(7)	6.3(2)	5.1(1)	16(24)	∞ *501*	0/15
DTS-C	**4.1**(0.9)	**3.3**(0.7)	**2.6**(0.8)	**2.3**(1)$^{\star 2}$	**1.9**(2)$^{\star 2}$	12/15
CMA-E	47(40)	37(33)	34(24)	120(194)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f10	*2.5e+6*:3.0	*6.3e+5*:7.0	*2.5e+5*:17	*6.3e+3*:54	*2.5e+1*:297	15/15
Ord-N	**2.6**(3)	2.9(3)	2.0(1)	3.0(0.9)	6.1(13)	4/15
Ord-H	4.1(3)	3.4(3)	1.7(1)	4.2(3)	12(17)	2/15
Ord-Q	2.7(3)	2.3(1)	1.9(1)	4.5(4)	25(10)	1/15
DTS-C	2.7(3)	**2.1**(2)	**1.2**(0.9)	**1.2**(0.5)$^{\star 2}$	**0.58**(0.1)$^{\star 4}$	15/15
CMA-E	3.7(9)	2.6(5)	2.1(2)	6.1(3)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f11	*1.0e+6*:3.0	*4.0e+4*:8.0	*6.3e+2*:16	*6.3e+1*:74	*6.3e-1*:298	15/15
Ord-N	2.9(3)	2.2(1)	4.3(2)	5.4(7)	∞ *502*	0/15
Ord-H	**2.0**(2)	**1.9**(2)	3.7(5)	3.3(5)	∞ *501*	0/15
Ord-Q	2.4(3)	3.0(2)	4.2(2)	3.1(0.8)	∞ *502*	0/15
DTS-C	2.7(3)	2.7(2)	**2.7**(0.9)	**0.93**(0.5)	**0.70**(0.1)$^{\star 4}$	15/15
CMA-E	3.7(6)	3.5(4)	11(6)	6.2(5)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f12	*4.0e+7*:4.0	*1.6e+7*:8.0	*4.0e+6*:19	*1.6e+4*:52	*1.0e+0*:268	15/15
Ord-N	2.8(3)	2.5(0.4)	**1.7**(0.6)	3.5(3)	∞ *501*	0/15
Ord-H	2.4(2)	**2.3**(2)	2.0(0.9)	3.0(0.7)	14(10)	2/15
Ord-Q	1.9(4)	2.6(1)	1.9(0.7)	3.2(1)	9.2(10)	3/15
DTS-C	**1.9**(2)	2.4(2)	2.1(1)	**1.9**(0.3)*	**1.5**(2)$^{\star 2}$	12/15
CMA-E	3.6(5)	3.7(3)	6.8(7)	144(178)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f13	*1.0e+3*:3.0	*6.3e+2*:8.0	*4.0e+2*:17	*6.3e+1*:52	*6.3e-2*:264	15/15
Ord-N	3.5(4)	2.6(0.9)	1.9(0.4)	2.5(5)	∞ *501*	0/15
Ord-H	3.4(3)	2.6(2)	1.9(0.5)	3.1(1)	∞ *501*	0/15
Ord-Q	4.2(3)	2.8(1)	1.7(0.4)	1.9(0.9)	28(25)	1/15
DTS-C	**2.5**(4)	**2.2**(2)	**1.5**(0.4)	**1.4**(0.1)	**1.4**(0.5)$^{\star 3}$	14/15
CMA-E	3.1(4)	3.8(2)	5.3(4)	13(4)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f14	*1.6e+1*:3.0	*1.0e+1*:10	*4.0e+0*:22	*2.5e-1*:53	*1.0e-5*:251	15/15
Ord-N	3.2(4)	1.6(1)	1.9(0.8)	2.9(2)	∞ *501*	0/15
Ord-H	**3.0**(3)	**1.2**(1)	1.7(1)	3.8(3)	∞ *501*	0/15
Ord-Q	4.1(4)	2.2(1)	1.6(0.9)	1.9(0.4)	∞ *501*	0/15
DTS-C	4.7(3)	1.8(1)	**1.5**(1.0)	**1.3**(0.3)*	**1.3**(0.1)$^{\star 4}$	15/15
CMA-E	6.2(9)	4.8(6)	5.9(2)	14(11)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f15	*1.6e+2*:3.0	*1.0e+2*:13	*6.3e+1*:24	*4.0e+1*:55	*1.6e+1*:289	5/5
Ord-N	4.8(3)	1.7(0.9)	1.5(1)	1.2(0.7)	8.0(9)	3/15
Ord-H	4.6(4)	**1.4**(1)	1.5(0.6)	1.4(0.5)	2.8(2)	7/15
Ord-Q	5.3(4)	1.7(1)	1.8(1)	1.6(0.7)	2.6(1.0)	8/15
DTS-C	**3.7**(4)	1.5(1)	**1.5**(0.4)	**1.0**(0.8)	**0.84**(0.5)	12/15
CMA-E	6.7(12)	3.3(3)	5.0(3)	5.5(5)	8.4(12)	3/15

#FEs/D	0.5	1.2	3	10	50	#succ
f16	*4.0e+1*:5.0	*2.5e+1*:16	*1.6e+1*:46	*1.0e+1*:120	*4.0e+0*:334	15/15
Ord-N	1.4(2)	**0.93**(0.9)	**1.3**(2)	1.9(2)	4.7(3)	4/15
Ord-H	**1.2**(1)	1.2(1)	1.8(1)	2.6(7)	22(13)	1/15
Ord-Q	1.2(2)	1.8(3)	2.4(2)	3.7(7)	22(19)	1/15
DTS-C	2.3(0.8)	1.6(0.8)	1.7(3)	**1.7**(3)	**2.5**(4)	6/15
CMA-E	1.9(2)	1.8(2)	2.1(2)	2.9(2)	22(34)	1/15

#FEs/D	0.5	1.2	3	10	50	#succ
f17	*1.0e+1*:5.0	*6.3e+0*:26	*4.0e+0*:57	*2.5e+0*:110	*6.3e-1*:412	15/15
Ord-N	3.5(3)	1.5(1)	**1.4**(1)	1.9(3)	4.1(4)	4/15
Ord-H	5.7(9)	2.0(0.5)	1.5(2)	1.4(2)	2.0(2)	7/15
Ord-Q	3.7(2)	**1.4**(0.9)	1.4(0.5)	**1.2**(1)	**1.4**(2)	9/15
DTS-C	**3.2**(5)	2.3(0.3)	1.4(2)	2.2(0.2)	3.5(7)	4/15
CMA-E	8.0(7)	3.6(2)	3.9(0.9)	3.1(1)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f18	*6.3e+1*:3.0	*4.0e+1*:7.0	*2.5e+1*:20	*1.6e+1*:58	*1.6e+0*:318	15/15
Ord-N	3.4(4)	2.9(4)	2.9(4)	1.6(0.7)	5.4(2)	4/15
Ord-H	2.9(4)	**2.8**(2)	**2.2**(2)	1.6(3)	11(14)	2/15
Ord-Q	**2.1**(3)	3.3(3)	2.5(2)	**1.6**(0.4)	7.0(9)	3/15
DTS-C	4.7(7)	8.0(20)	5.2(7)	2.1(3)	**1.7**(2)	8/15
CMA-E	5.9(15)	4.2(8)	3.6(1)	2.7(2)	23(24)	1/15

#FEs/D	0.5	1.2	3	10	50	#succ
f19	*1.6e-1*:172	*1.0e-1*:242	*6.3e-2*:675	*4.0e-2*:3078	*2.5e-2*:4946	15/15
Ord-N	∞	∞	∞	∞	∞ *502*	0/15
Ord-H	∞	∞	∞	∞	∞ *502*	0/15
Ord-Q	∞	∞	∞	∞	∞ *502*	0/15
DTS-C	∞	∞	∞	∞	∞ *500*	0/15
CMA-E	∞	∞	∞	∞	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f20	*6.3e+3*:5.0	*4.0e+3*:8.0	*2.5e+1*:16	*2.5e+0*:69	*1.0e+0*:851	15/15
Ord-N	**2.4**(2)	2.0(2)	2.6(1.0)	3.9(2)	∞ *501*	0/15
Ord-H	2.7(3)	2.4(2)	2.6(1)	3.1(2)	∞ *501*	0/15
Ord-Q	2.7(3)	2.3(1)	2.7(0.8)	**2.8**(2)	∞ *501*	0/15
DTS-C	2.5(2)	**1.8**(2)	**1.7**(0.6)	2.9(3)	∞ *500*	0/15
CMA-E	4.3(4)	3.5(3)	8.6(6)	8.0(6)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f21	*4.0e+1*:4.0	*2.5e+1*:11	*1.6e+1*:31	*6.3e+0*:73	*1.6e+0*:347	5/5
Ord-N	**1.6**(2)	2.0(2)	2.2(2)	4.8(4)	10(9)	2/15
Ord-H	3.1(3)	2.4(2)	3.1(2)	5.3(10)	4.7(5)	4/15
Ord-Q	2.4(4)	2.7(2)	2.5(2)	7.6(8)	**4.6**(10)	4/15
DTS-C	2.2(2)	**1.5**(1)	**1.0**(0.6)	**2.5**(4)	10(9)	2/15
CMA-E	1.9(1)	1.5(2)	2.1(5)	4.1(1)	∞ *500*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f22	*6.3e+1*:4.0	*4.0e+1*:15	*2.5e+1*:32	*1.0e+1*:71	*1.6e+0*:341	5/5
Ord-N	**1.4**(0.9)	**1.3**(2)	**1.4**(1)	**2.4**(2)	10(13)	2/15
Ord-H	2.4(3)	1.8(2)	2.7(1)	3.5(6)	**3.4**(4)	5/15
Ord-Q	3.1(3)	2.4(2)	2.7(1)	3.2(6)	4.5(7)	4/15
DTS-C	2.8(4)	3.6(9)	3.3(12)	5.3(4)	4.2(5)	4/15
CMA-E	2.0(4)	2.1(3)	1.5(1)	4.2(2)	7.1(4)	3/15

#FEs/D	0.5	1.2	3	10	50	#succ
f23	*1.0e+1*:3.0	*6.3e+0*:9.0	*4.0e+0*:33	*2.5e+0*:84	*1.0e+0*:518	15/15
Ord-N	2.6(3)	3.2(2)	2.6(0.7)	8.9(6)	∞ *502*	0/15
Ord-H	**2.0**(3)	2.3(3)	3.0(2)	7.2(5)	∞ *502*	0/15
Ord-Q	2.6(2)	3.7(3)	2.8(3)	6.6(12)	∞ *501*	0/15
DTS-C	2.1(2)	**2.0**(2)	3.0(4)	**1.5**(6)	**0.94**(1)*	9/15
CMA-E	2.2(1)	3.3(6)	**2.4**(0.8)	5.6(4)	14(9)	1/15

#FEs/D	0.5	1.2	3	10	50	#succ
f24	*6.3e+1*:15	*4.0e+1*:37	*4.0e+1*:37	*2.5e+1*:118	*1.6e+1*:692	15/15
Ord-N	2.1(1)	2.2(2)	2.2(3)	2.5(2)	11(10)	1/15
Ord-H	1.8(1)	2.0(0.4)	2.0(0.8)	4.9(3)	2.5(2)	4/15
Ord-Q	2.1(1)	**1.5**(1)	**1.5**(1)	2.9(5)	10(17)	1/15
DTS-C	**1.7**(1)	3.0(7)	3.0(7)	**2.0**(5)	**0.84**(2)	8/15
CMA-E	4.6(4)	5.8(3)	5.8(3)	5.8(8)	3.6(4)	3/15

Table 4: Average runtime (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2009 in dimension 5. The aRT and in braces, as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and run-length based target, the corresponding reference aRT (preceded by the target Δf-value in *italics*) in the first row. #succ is the number of trials that reached the target value of the last column. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction of 110. A ↓ indicates the same tested against the best algorithm from BBOB 2009. Best results are printed in bold.

Data produced with COCO v2.0.1