ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 163–171
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 J. Repický, L. Bajer, M. Holeňa

Traditional Gaussian Process Surrogates in the BBOB Framework

Jakub Repický[1], Lukáš Bajer[1], and Martin Holeňa[2]

[1] Faculty of Mathematics and Physics
Charles University in Prague
Malostranské nám. 25
Prague, Czech Republic
{j.repicky,bajeluk}@gmail.com
[2] Institute of Computer Science,
Czech Academy of Sciences
Pod Vodárenskou věží 2
Prague, Czech Republic
martin@cs.cas.cz

Abstract: Objective function evaluation in continuous optimization tasks is often the operation that dominates the algorithm's cost. In particular in the case of black-box functions, i.e. when no analytical description is available, and the function is evaluated empirically. In such a situation, utilizing information from a surrogate model of the objective function is a well known technique to accelerate the search. In this paper, we review two traditional approaches to surrogate modelling based on Gaussian processes that we have newly reimplemented in MATLAB: Metamodel Assisted Evolution Strategy using probability of improvement and Gaussian Process Optimization Procedure. In the research reported in this paper, both approaches have been for the first time evaluated on Black-Box Optimization Benchmarking framework (BBOB), a comprehensive benchmark for continuous optimizers.

1 Introduction

An analytical definition of the objective function in real-world optimization tasks is sometimes hard to obtain. Therefore, neither information about the function's smoothness nor its derivatives are available. Moreover, evaluation of the function is usually expensive as it can only be done empirically, e.g. by measurement, testing, or running a computer simulation. Such functions are called black-box.

One class of optimization algorithms that are successfully applied to black-box optimization are evolution strategies. An evolution strategy is an optimization method that works on a population of candidate solutions using evolutionary operators of selection, mutation and recombination [10] [11]. In particular, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [4] is considered to be the state-of-the-art continuous black-box optimizer. It samples each population according to a multivariate normal distribution determined by a covariance matrix. A notable property of the algorithm is the adaptation of the covariance matrix along the path of past successful search steps.

Still the whole population must be evaluated which might make running the algorithm for a sufficient number of generations infeasible due to expensive evaluation. To address this issue, techniques that involve a surrogate regression model of the fitness have been proposed.

Two major requirements of incorporating a surrogate model (also called metamodel in the literature) into evolutionary strategies are model management and model selection.

Model management is the task to control the surrogate model's impact on algorithm's convergence by using the original fitness alongside its surrogate model in the course of the search.

In *evolution control*, a certain fraction of individuals or generations is controlled, i.e. evaluated with the fitness function, while the remainder is evaluated with the surrogate model [8].

For example, *Metamodel-Assisted Evolution Strategy* (MAES) uses a surrogate model to pre-select the most promising individuals before they enter a selection procedure of a standard ES [3].

In contrast to evolution control, *surrogate approach* [2] directly optimizes the model output in an iterative procedure, thus avoiding the issue of determining the correct fraction of controlled individuals. In each iteration, a fixed number of candidate solutions are found by minimizing the model with an evolution strategy. These solutions are thereafter evaluated on the real fitness and the model is updated.

Regarding the model selection, Gaussian processes (GPs) are a non-parameterized regression model that is appealing for the task as it gives its prediction in terms of a Gaussian distribution. The variance of this prediction can be utilized as a confidence measure that promotes exploration of insufficiently modelled areas.

This paper reviews two traditional algorithms interconnecting Gaussian process-based surrogate models with the CMA-ES: Metamodel-Assisted Evolution strategy with improved pre-selection criterion by Ulmer, Strechert and Zell [13] and Gaussian Process Optimization Procedure (GPOP) by Büche, Schraudolph and Koumoutsakos [2].

The former is a GP-based MAES with probability of improvement (POI) as a pre-selection criterion.

The latter represents the surrogate approach – in each iteration, a local GP model is built and four functions designed to balance predicted value and variance are optimized.

While both algorithms are GP-based, they differ both in the model-management approach as well as in utilizing GP's confidence.

The framework COCO/BBOB (Comparing Continuous Optimizers / Black-Box Optimization Benchmarking) [7] provides an experimental setup for benchmarking black-box optimizers. In particular, its noiseless testbed [6] comprises of 24 functions with properties that are to different extents challenging for continuous optimizers.

Both tested methods had been proposed before the BBOB framework originated. In the research reported in this paper, we evaluated both methods on the noiseless part of the BBOB framework. For that purpose a new implementation[1] was required as the original source codes were not available to us.

In the following, we first briefly introduce Gaussian processes as a suitable surrogate fitness model in Section 2. An exposition of the tested methods is given in Section 3 and experimental setup in Section 4. Section 5 presents experimental results and finally Section 6 concludes the paper.

2 Gaussian processes

Both algorithms under review feature the choice of Gaussian processes as a surrogate fitness function model.

GPs are a probabilistic model with several properties that make it well suited for fitness function modelling: its hyperparameters are comprehensible and limited in number and it provides a confidence measure given by standard deviation of predicted value at new data points.

In the following, we define a GP using notation and equations from Büche [2].

Consider $f\colon \mathbb{R}^D \to \mathbb{R}$ an unknown real-parameter function to be approximated. GP model is specified by a set $\mathbf{X}_N = \left\{x_i \,|\, x_i \in \mathbb{R}^D\right\}_{i=1}^N$ of N training data points with known function values $\mathbf{t}_N = \{t_i \,|\, f(\mathbf{x}_i) = t_i\}_{i=1}^N$. The data are assumed to be a sample of zero-mean multivariate Gaussian distribution with joint probability density

$$p(\mathbf{t}_N \,|\, \mathbf{X}_N) = \frac{\exp(-\frac{1}{2}\mathbf{t}_N^T \mathbf{C}_N^{-1} \mathbf{t}_N)}{\sqrt{(2\pi)^N \det(\mathbf{C}_N)}} \tag{1}$$

where the covariance matrix $\mathbf{C}_N$ is defined by means of a covariance function $C(\mathbf{x}_i, \mathbf{x}_j, \Theta)$ $i, j \in \{1, \ldots, N\}$ with a fixed set of hyperparameters Θ.

[1]The sources are available at `https://github.com/repjak/surrogate-cmaes/`

For a set $\mathbf{t}_{N+1}$ that includes a new observation $t_{N+1} = f(\mathbf{x}_{N+1})$, we obtain

$$p(\mathbf{t}_{N+1} \,|\, \mathbf{X}_{N+1}) = \frac{\exp(-\frac{1}{2}\mathbf{t}_{N+1}^T \mathbf{C}_{N+1}^{-1} \mathbf{t}_{N+1})}{\sqrt{(2\pi)^{N+1} \det(\mathbf{C}_{N+1})}}. \tag{2}$$

Using Bayesian rule for conditional probabilities, the prediction at a new data point has the density function

$$p(t_{N+1} \,|\, \mathbf{X}_{N+1}, \mathbf{t}_N) = \frac{p(\mathbf{t}_{N+1} \,|\, \mathbf{X}_{N+1})}{p(\mathbf{t}_N \,|\, \mathbf{X}_N)}. \tag{3}$$

The covariance matrix $\mathbf{C}_{N+1}$ can be written with the use of the covariance matrix $\mathbf{C}_N$ as

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{pmatrix} \tag{4}$$

where $\mathbf{k} = (C(\mathbf{x}_i, \mathbf{x}_{N+1}))_1^N$ is a vector of covariances between the new point and $\mathbf{X}_N$ and $\kappa = C(\mathbf{x}_{N+1}, \mathbf{x}_{N+1})$ is the new point's variance.

Using (1) and (2) together with the fact that the inverse $\mathbf{C}_{N+1}^{-1}$ can also be expressed by the means of $\mathbf{C}_N^{-1}$, (3) can be simplified to a univariate Gaussian [2]

$$p(t_{N+1} \,|\, \mathbf{X}_{N+1}, \mathbf{t}_N) \propto \exp\left(-\frac{1}{2}\frac{(t_{N+1} - \hat{t}_{N+1})^2}{\sigma_{t_{N+1}}^2}\right) \tag{5}$$

with mean and variance given by

$$\begin{aligned} \hat{t}_{N+1} &= \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N, \\ \sigma_{t_{N+1}}^2 &= \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}. \end{aligned}$$

A comprehensive exposition of Gaussian processes can be found in [9].

The covariance function plays an important role as it expresses prior assumptions about the shape of the modelled function. The vector Θ of model's hyperparameters is usually fitted with the maximum likelihood method.

One of the most commonly used covariance functions is squared exponential covariance function:

$$C(\mathbf{x}_p, \mathbf{x}_q) = \theta exp\left(-\frac{1}{2}\frac{(\mathbf{x}_p - \mathbf{x}_q)^T (\mathbf{x}_p - \mathbf{x}_q)}{r^2}\right) \tag{6}$$

where the parameter θ scales the covariance between two points and radius r is the characteristic length scale of the process.

When two points are close to each other compared to the characteristic length scale r, the covariance is close to one while it exponentially decays to zero with their distance growing.

The squared exponential covariance function can be improved with automatic relevance determination (ARD):

$$C(\mathbf{x}_p, \mathbf{x}_q) = \theta exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_{p,i} - x_{q,i})^2}{r_i^2}\right) \tag{7}$$

where the radii r_i scale the impact of two points distance on their correlation separately in each dimension.

To address the need to balance the exploitation of a fitted model prediction and the exploration of regions where model's confidence is low, the standard deviation of GP's prediction can be utilized.

One possibility are *merit functions* proposed in [12] for the purpose of balancing exploration and exploitation in engineering design optimization. Consider M to be a trained Gaussian process model. A merit function combines the goals of finding a minimum predicted by the model M and improving the accuracy of M into a single objective function:

$$f_{M,\alpha}(\mathbf{x}) = \hat{t}(\mathbf{x}) - \alpha\sigma(\mathbf{x}) \tag{8}$$

where $\hat{t}(\mathbf{x})$ is the mean of the prediction of the function value in $\mathbf{x}$, $\sigma(\mathbf{x})$ is the prediction standard deviation and $\alpha \geq 0$ is a balancing parameter.

Another option is the probability of improvement (POI). Let us assume Gaussian process prediction to be a random variable $Y(\mathbf{x})$ with mean $\hat{t}(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$. For a chosen threshold T less than or equal to the so-far best obtained fitness value $f_{\min}$, the probability of improvement in point $\mathbf{x}$ is defined as:

$$POI_T(\mathbf{x}) = p(Y(\mathbf{x}) \leq T) = \Phi\left(\frac{T - \hat{t}(\mathbf{x})}{\sigma(\mathbf{x})}\right) \tag{9}$$

where Φ is the cumulative distribution function of the distribution $\mathcal{N}(0,1)$.

3 Tested Methods

3.1 GP Model Assisted Evolution Control

A standard ES operates on λ offsprings generated from μ parents by evolutionary operators of recombination and mutation. After the fitness of the offsprings is evaluated, a population of μ best individuals is selected to reproduce to a new offspring in the next iteration. In (μ,λ) ES, μ best individuals are selected from the λ offsprings, whereas in $(\mu+\lambda)$ ES, μ best individuals are selected from the union of the offprings and their parents.

MAES [3] modifies the standard evolution strategy with producing $\lambda_{\text{Pre}} > \lambda$ instead of λ individuals from μ parents by the same operators of recombination and mutation (steps 4 and 5 in the Algorithm 1). Given a fitted Gaussian process M, individuals $\mathbf{x}_i$, $i = 1,\ldots,\lambda_{\text{Pre}}$ are then preselected according to a criterion χ_M defined for the model M to create λ offsprings (step 6).

The GP model is trained in every generation on a set of N_{tr} most recently evaluated individuals (step 8).

In this paper we consider two pre-selection criteria in accordance with [13]: mean model prediction (MMP) (5) which selects λ_{Pre} points with the best mean predicted fitness $\hat{t}(\mathbf{x})$ and POI (9). The authors of [13] prefer POI to merit functions (8), as it does not depend on finding the appropriate value of the scale parameter α.

Algorithm 1 GP Model-Assisted Evolution Strategy

```
Input: $f$ – fitness function
       $\mu$ – number of parents
       $\lambda$ – population size
       $\lambda_{\text{Pre}}$ – size of pre-selected population
       $N_{\text{tr}}$ – size of training dataset
       $\chi_M$ – the preselection criterion that depends on a GP
       model $M$, e.g. mean model prediction or POI
 1: Pop $\leftarrow$ generate and evaluate $\lambda$ initial samples
 2: $M \leftarrow$ a GP model trained on points from Pop
 3: while termination criteria not reached do
 4:     Offspring $\leftarrow$ reproduce Pop into $\lambda_{\text{Pre}}$ new points
 5:     Offspring $\leftarrow$ mutate Offspring
 6:     Offspring $\leftarrow$ select best $\lambda$ points according to the
        pre-selection criterion $\chi_M$
 7:     evaluate Offspring with $f$
 8:     $M \leftarrow$ update model $M$ on $N_{\text{tr}}$ points most recently
        evaluated with $f$
 9:     Pop $\leftarrow$ select $\mu$ points best according to $f$
10: end while
```

3.2 Gaussian Process Optimization Procedure

Gaussian Process Optimization Procedure is due to Büche, Schraudolph and P. Koumoutsakos [2]. A Gaussian process is trained on a subset of already evaluated data and optimized by an ES instead of the original fitness. As optimization of the surrogate is cheaper than optimization of the original fitness, this can be repeated until reaching some termination criteria.

CMA-ES is used as the evolution strategy, with the number of parents μ set to 2 and the population size λ set to 10.

The pseudocode is given in Figure 2. After generating an initial population, a local GP model is built and utilized in an iterative procedure. Considering possibly low accuracy and computational infeasibility of global models, the training dataset is restricted to N_C points closest to the current best known solution $\mathbf{x}^{\text{best}}$ (step 6) and N_R most recently evaluated points (step 7).

If the GP model M has been successfully trained then the CMA-ES optimizes four merit functions $f_{M,\alpha}$ (8) for each $\alpha \in \{0,1,2,4\}$. Areas that might be approximated inaccurately are avoided by bounding the ES to the hypercube spanning the set of N_C points selected from $\mathbf{x}^{\text{best}}$'s neighborhood (steps 10 and 12). The points that are optima of the considered merit functions are evaluated by the original fitness and added to the dataset of known points.

In the case that no new solution is found, a random perturbation (step 23) is evaluated and added to the dataset. Unfortunately, authors don't specify the parameter m that occurs in 23. We set it to the value $m = 1$.

The authors used the following covariance function in their GP model:

$$C(\mathbf{x}_p, \mathbf{x}_q) = \theta_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\frac{(x_{p,i}-x_{q,i})^2}{r_i^2}\right) + \theta_2 + \delta_{pq}\theta_3 \tag{10}$$

where r_i, $\theta_1, \theta_2, \theta_3 > 0$ and δ_{pq} is the Kronecker delta. The function is the sum of the squared exponential covariance function with ARD (7), constant shift θ_2 and a white noise scaled with θ_3.

Algorithm 2 Gaussian Process Optimization Procedure

```
Input: $N_C$ – number of training points selected according
         to their distance to $\mathbf{x}^{\text{best}}$
       $N_R$ – number of training points selected according to
         most recent time of evaluation
       $f$ – fitness function
       $\mu$ – the number of parents for CMA-ES
       $\lambda$ – population size for CMA-ES
 1: $\{\mathbf{x}_1, \dots, \mathbf{x}_{N_C/2}\} \leftarrow$ a set of $N_C/2$ points generated by CMA-ES
 2: $y_i \leftarrow f(\mathbf{x}_i) \quad i \in \{1, \dots, N_C/2\}$
 3: $A \leftarrow \{(\mathbf{x}_i, y_i) \mid i \in \{1, \dots, N_C/2\}\}$
 4: $\mathbf{x}^{\text{best}} \leftarrow \arg\min_{\mathbf{x} \in \{\mathbf{x}_i \mid i \in \{1,\dots,N_C/2\}\}} (f(\mathbf{x}))$
 5: while stopping criteria not reached do
 6:   $T_C \leftarrow N_C$ points from $\{\mathbf{u} \mid \exists z (\mathbf{u}, z) \in A\}$ closest to $\mathbf{x}^{\text{best}}$
 7:   $T_R \leftarrow N_R$ points most recently evaluated
 8:   $M \leftarrow$ a GP model trained on $T_C \cup T_R$
 9:   $S \leftarrow \emptyset$
10:   $\mathbf{d} \leftarrow (d_i)_{i=1}^{D}$, $d_i = \max_{\mathbf{x} \in T_C}(x_i) - \min_{\mathbf{x} \in T_C}(x_i)$
11:   for all $\alpha \in \{0, 1, 2, 4\}$ do
12:     $B \leftarrow \left\{\mathbf{x} \mid \mathbf{x}^{\text{best}} - \frac{\mathbf{d}}{2} \leq \mathbf{x} \leq \mathbf{x}^{\text{best}} + \frac{\mathbf{d}}{2}\right\}$
13:     $\mathbf{x} \leftarrow$ optimize $f_{M,\alpha}$ within $B$ by $(\mu, \lambda)$-CMA-ES
14:     if $\nexists z\, (\mathbf{x}, z) \in A$ then
15:       $y \leftarrow f(\mathbf{x})$
16:       $S \leftarrow S \cup \{\mathbf{x}, y\}$
17:     end if
18:   end for
19:   if $S \neq \emptyset$ then
20:     $A \leftarrow A \cup S$
21:     $\mathbf{x}^{\text{best}} \leftarrow \arg\min_{\mathbf{x} \in \{\mathbf{u} \mid \exists z (\mathbf{u}, z) \in A\}} (f(\mathbf{x}))$
22:   else
23:     $\mathbf{x}^R \leftarrow \left(x_i^{\text{best}} + \frac{z_i d_i}{100} m\right)_{i=1}^{D} \quad z_i \sim \mathcal{N}(0, 1)$
24:     $y^R \leftarrow f(\mathbf{x})$
25:     $A \leftarrow A \cup \left\{(\mathbf{x}^R, y^R)\right\}$
26:   end if
27: end while
28: return $\mathbf{x}^{\text{best}}$
```

4 Experimental Setup

The framework COCO/BBOB (Comparing Continuous Optimizers / Black-Box Optimization Benchmarking) [6] [7] is intended for systematic experimentation with black-box optimizers. We use the noiseless testbed of the BBOB framework, which is comprised of 24 real-parameter benchmark functions with different characteristics such as non-separability, multi-modality, ill-conditioning etc.

Each function is defined on $[-5, 5]^D$ for all $D \geq 2$. For every function and every dimensionality, 15 trials of the optimizer are run. A different instance of the original function is used for each trial. We used dimensionalities $2, 3, 5, 10$, thus 1440 trials in total were run for each set of parameters and each method.

Since source codes were available for neither of the tested methods, we implemented them in MATLAB.

For Gaussian processes, we chose MATLAB's default implementation, `fitrgp`, a part of Statistics and Machine Learning Toolbox. The GP hyperparameters fitting was done with a quasi-newton optimizer, which is the default in `fitrgp`.

Parameters of the benchmarked algorithms were set as follows:

CMA-ES. A multi-start version with the population size doubled after each restart was used in the tests (MATLAB code v. 3.62.beta). Number of restarts was set to 4, while other parameters settings were left default: $\lambda = 4 + \lfloor 3\log_{10} D \rfloor$, $\sigma_{\text{start}} = \frac{8}{3}$, IncPopSize $= \lfloor \lambda/2 \rfloor$.

MAES. We implemented GP Model Assisted Evolution Strategy on top of a framework developed for S-CMA-ES algorithm, which employs a GP model in conjunction with a generation evolution control [1]. The S-CMA-ES implementation allows to conveniently replace the population sampling step of the CMA-ES with a custom procedure. In this case, a population intended for pre-selection is sampled and processed as described in Subsection 3.1. The control is then handed over back to the CMA-ES.

The number of parents and population size were set to correspond with the CMA-ES settings.

Two pre-selection criteria were tested: MMP and the POI with threshold equal to the so-far best sampled fitness value $f_{\min}$. In both cases λ_{Pre} was set to 3λ. The training set was comprised from 2λ most recently evaluated points.

We used the same covariance function as in the GPOP case, i.e. (10).

GPOP. We adhered to [2] in usage of the proposed covariance function (10).

The termination criteria were chosen as follows:

- number of consecutive iterations with no new solution found is larger than 2 while the tolerance on the two points euclidean distance for them to be considered equal is 10^{-8}
- the overall change of fitness values during the last 10 iterations is lower than 10^{-9}

- the target fitness value is reached

Training set size parameters N_C and N_R were chosen in accordance with empirical results reported in [2], particularly $N_C = N_R = 5 * D$.

Although the performance is measured in terms of the number of fitness evaluations, other operations such as optimizing a surrogate model may also be costly in benchmarking scenarios. If we consider functions in 10 D, a run of GPOP on one core of a computational grid took approximately 27.8 real hours per function on average. For those reasons, we limited the maximum number of fitness evaluations to $100 * D$ for all tested methods.

5 Experimental Results

Results from experiments on all the 24 noiseless BBOB benchmark functions are presented in Figures 1–3.

The expected running time (ERT) depicted in Figure 1 depends on a given target value, $f_t = f_{opt} + \Delta f$, i.e. the true optimum f_{opt} of the respective benchmark function raised by a small value Δf. The ERT is computed over all relevant trials as the number of the original function evaluations (FE) executed during each trial until the target function value f_t reached, summed over all trials and divided by the number of successful trials, i.e. trials that actually reached f_t: [7]

$$\mathrm{ERT}(f_t) = \frac{\#\mathrm{FE}(f_{\mathrm{best}} \geq f_t)}{\#\mathrm{succ}} \tag{11}$$

In the graphs for functions 1, 2, 5, 8, 9, 12, 14, GPOP achieved significantly better results compared to all other algorithms for some dimensions. In contrast, the differences between MAES and CMA-ES are rather small, regardless of the pre-selection criterion.

The graphs in Figure 2 summarize the performance over subgroups of the benchmark functions for the highest tested dimension 10. The graphs show the proportion of algorithm runs that reached a target value $f_t \in 10^{[-1..2]}$ (see the figure caption for further details).

The GPOP speedup is most eminent on the group of separable functions (functions 1–5), that is functions, optimization of which can be reduced to D one-dimensional problems [6]. On the other hand, GPOP has the worst results of all methods on multi-modal functions (functions 15–19).

Similar results may be observed on Figure 3 that for each function shows the dependence of the relative best fitness value on the number of evaluations in 10 D. As can be seen on the graphs for functions 13 and 24, GPOP in some cases outperforms all other algorithms in early stages, but then gets trapped in a local minimum.

On the graph for functions 21 on Figure 3, MAES with MMP as the pre-selection criterion visibly outperforms all other algorithms.

6 Conclusion

In this paper, we compared the CMA-ES and our implementation of two traditional methods which improve upon the CMA-ES with Gaussian process-based surrogate models: MAES, which extends the CMA-ES with a preselection step, and GPOP, which iteratively optimizes the GP model.

The benchmarks on the BBOB framework did not show any significant speedup of MAES compared to the CMA-ES. On the other hand, GPOP in many cases outperforms all the other methods, especially in early optimization stages. On some functions, though, it tends to get trapped in local minima. This might be explained with the fact that GPOP requires considerably fewer function evaluations per iteration than other methods. However, the model is built locally and might not be sufficient for exploration of the search space in later phases.

Our MAES implementation relies on a modification of the sampling step in CMA-ES, thereby changing the distribution of the sampled population. This might mislead CMA-ES a bit and requires further research, in particular considering the proposal in [5].

Another area for further research is the exploration of various confidence measures across tested methods, especially in connection with GPOP.

7 Acknowledgments

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

The research performed by Lukáš Bajer has been supported by the Czech Health Research Council project NV15-33250A.

The research performed by Jakub Repický was supported by SVV project number 260 224.

References

[1] L. Bajer, Z. Pitra, and M. Holeňa. Benchmarking Gaussian processes and random forests surrogate models on the BBOB noiseless testbed. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, 2015.

[2] D. Büche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics*, 35(2):183–194, 2005.

[3] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *In Parallel Problem Solving from Nature VII*, pages 361–370. Springer, 2002.

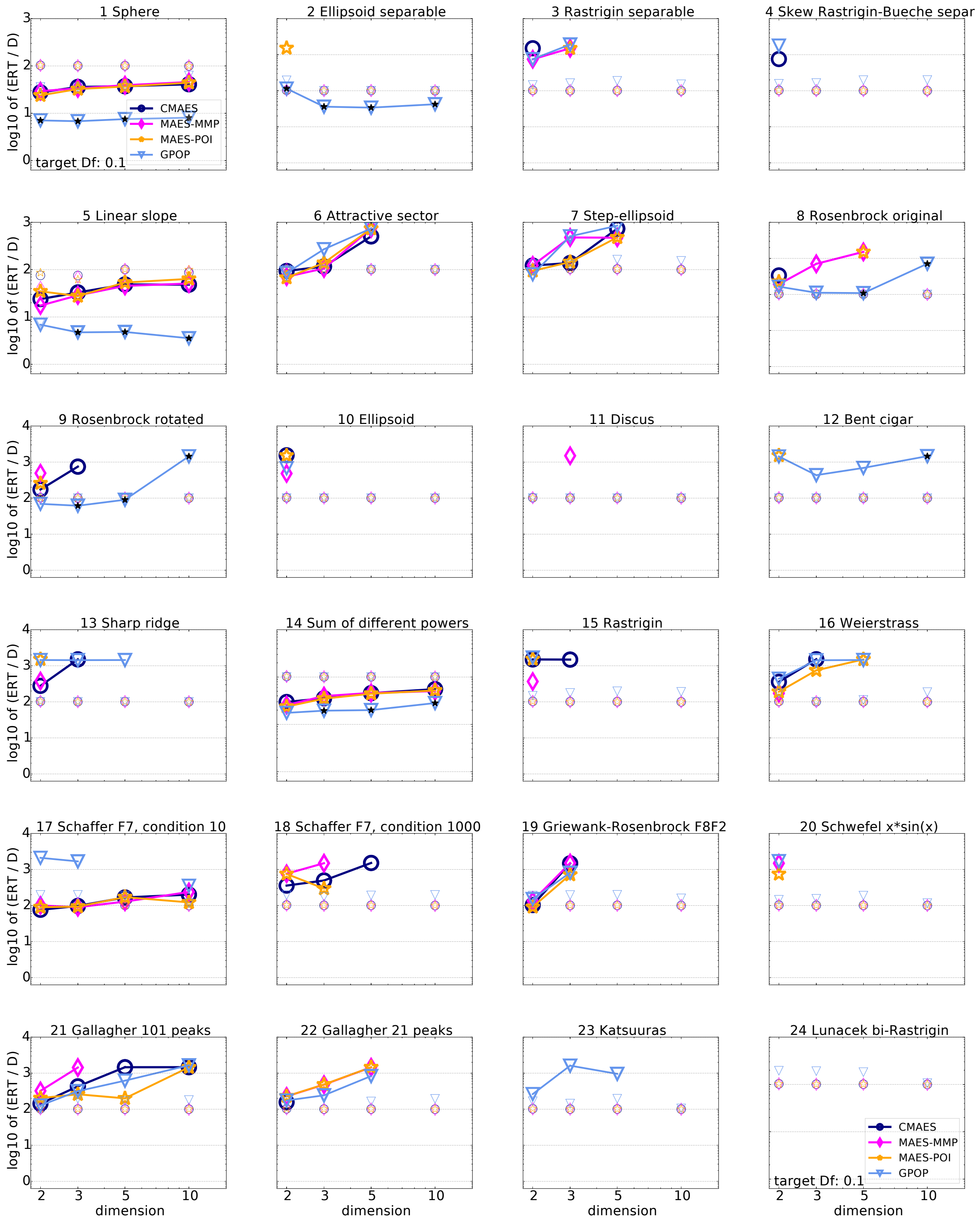

Figure 1: Expected running time (ERT as $\log_{10}$ of the number of f-evaluations) for target function value 0.1 divided by dimension, versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Values are plotted up to the highest dimensionality with at least one successful trial. 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 $p < 0.05$ using Hommel correction.

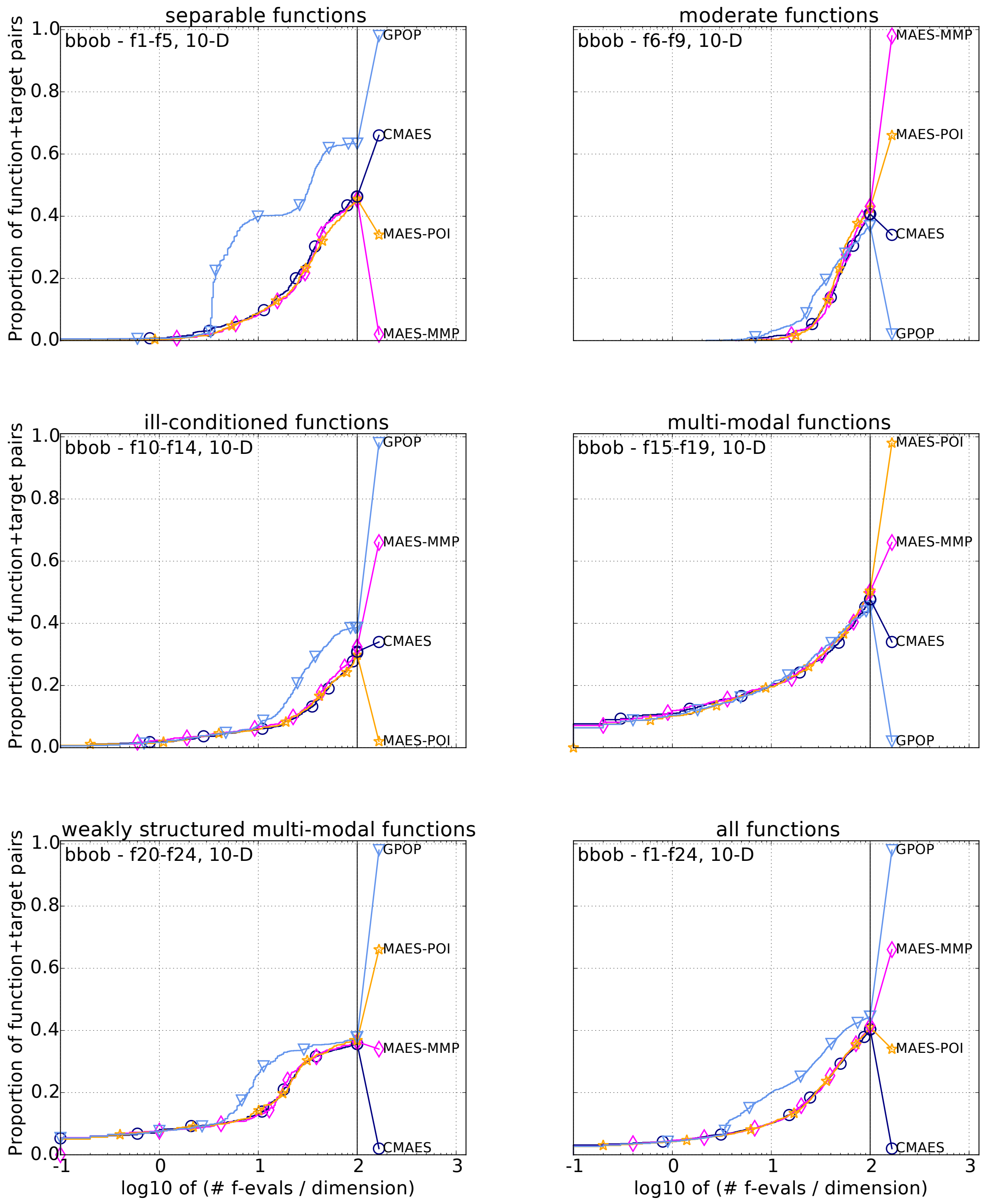

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FE/D) for 61 targets with target precision in $10^{[-1..2]}$ for different subgroups of BBOB benchmark functions in 10 D.

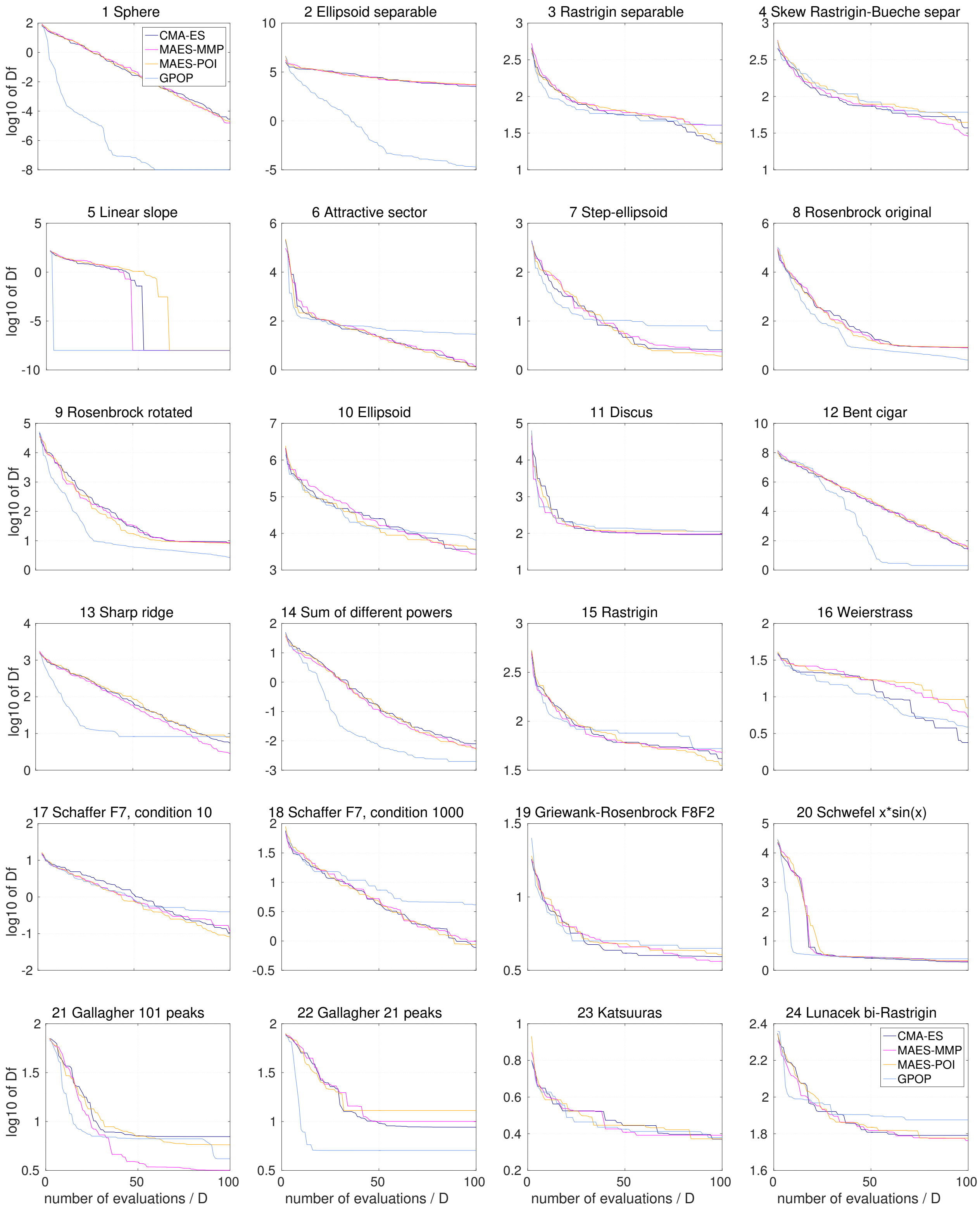

Figure 3: Relative fitness of best individual as $\log_{10}$ value versus number of evaluations divided by dimension for all functions in 10 D. Best fitness value at each number of evaluations is computed as the median of all trials.

[4] N. Hansen. The CMA evolution strategy: a comparing review. In I. Inza J. A. Lozano, P. Larrañaga and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[5] N. Hansen. Injecting external solutions into CMA-ES. *CoRR*, abs/1110.4181, 2011.

[6] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. technical report rr-6829. Technical report, INRIA, 2009. Updated February 2010.

[7] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2012: Experimental setup. technical report. Technical report, INRIA, 2012.

[8] Y. Jin and B. Sendhoff. Fitness approximation in evolutionary computation-a survey. In *GECCO 2002 Proceedings of Genetic and Evolutionary Computation Conference*, pages 1105–1111, 2002.

[9] C. E. Rassmusen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning series. MIT Press, 2006.

[10] I. Rechenberg. *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.

[11] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen*. Birkhäuser, Basel, 1977.

[12] V. Torczon and M. W. Trosset. Using approximations to accelerate engineering design optimization. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis & Optimization Symposium (Held at Saint Louis, Missouri), paper 98-4800*, pages 507–512, 1998.

[13] H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by Gaussian processes with improved pre-selection criterion. *IEEE Congress on Evolutionary Computation*, pages 692–699, 2003.