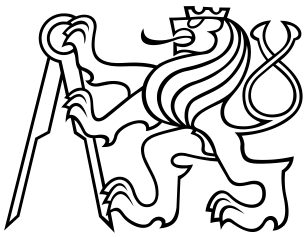
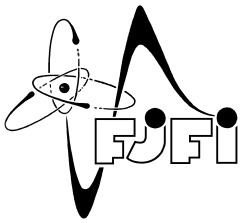
CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF NUCLEAR SCIENCES AND PHYSICAL ENGINEERING



DOCTORAL THESIS

SURROGATE MODELING AND LANDSCAPE ANALYSIS FOR EVOLUTIONARY BLACK-BOX OPTIMIZATION



PRAGUE 2022

ZBYNĚK PITRA

DECLARATION

I declare that I carried out this Thesis independently and only with the cited sources, literature and other professional sources.

Prague, March 2022

Zbyněk Pitra

BIBLIOGRAFICKÝ ZÁZNAM

Autor	Ing. Zbyněk Pitra České vysoké učení technické v Praze Fakulta jaderná a fyzikálně inženýrská Katedra matematiky
Název práce	Náhradní modelování a analýza tvaru funkcí v evoluční black-box optimalizaci
Studijní obor	Matematické inženýrství
Školitel	prof. RNDr. Ing. Martin Holeňa, CSc. Ústav informatiky, v.v.i. Akademie věd České republiky
Akademický rok	2022
Počet stran	190
Klíčová slova	náhradní modelování, analýza tvaru funkcí, black-box optimalizace, evoluční strategie, evoluční kontrola

BIBLIOGRAPHIC ENTRY

Author	Ing. Zbyněk Pitra Czech Technical University in Prague Faculty of Nuclear Sciences and Physical Engineering Department of Mathematics
Title of Dissertation	Surrogate Modeling and Landscape Analysis for Evolutionary Black-box Optimization
Field of Study	Mathematical Engineering
Supervisor	prof. RNDr. Ing. Martin Holeňa, CSc. Institute of Computer Sciences Czech Academy of Sciences
Academic Year	2022
Number of Pages	190
Keywords	surrogate modelling, landscape analysis, black-box optimization, evolutionary strategies, evolution control

This thesis is dedicated to my beloved Kamča and Emilka.

Zbyněk Pitra: *Surrogate Modeling and Landscape Analysis for Evolutionary Black-box Optimization,*
Doctoral Thesis, © March 2022

ABSTRAKT

Tato práce se zabývá dvěma oblastmi zabývající se statickými modely schopnými urychlit optimalizaci reálných cílových funkcí drahých ať už finančně nebo z hlediska časové náročnosti — evolučními algoritmy v kombinaci s regresními náhradními modely a analýzou tvaru funkcí pro náhradní modely v kontextu evoluční black-box optimalizace.

První část práce se soustředí na využití náhradních modelů schopných předpovídat rozdělení celé cílové black-box funkce v kombinaci s evolučním algoritmem považovaným za jeden z nejlepších v oblasti black-box optimalizace: Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Z tohoto výzkumu, na který se soustředí hlavní část této práce, vznikl Doubly Trained Surrogate CMA-ES (DTS-CMA-ES). Tento algoritmus využívá nejistotu předpovědi gaussovského procesu k výběru části populace CMA-ESu, kterou ohodnotí drahou cílovou funkcí, zatímco jeho střední hodnotu použije k ohodnocení zbytku populace. DTS-CMA-ES překonává další velmi dobré optimalizační algoritmy s náhradními modely v řadě benchmarkových testů.

Druhá část práce se zabývá vztahy mezi přesností regrese náhradních modelů a příznaky popisujícími tvar zkoumané black-box funkce. Tato část také studuje vlastnosti těchto příznaků v kontextu různých transformací dat a způsobů jejich výběru. Analýza tvaru funkcí je provedena na několika různých datasetech vygenerovaných z bezšumových funkcí integrovaných v systému COmparing Continuous Optimizers.

ABSTRACT

The thesis covers two areas concerning statistical models able to save resources or to speed-up the optimization of an expensive or time-consuming real-world objective function — evolutionary computation assisted by regresion surrogate models and landscape analysis of surrogate models in evolutionary black-box context.

The first part focuses on combining surrogate models capable to predict the whole distribution of the optimized function with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), the state-of-the-art evolutionary algorithm in continuous black-box optimization. Such combination, described in the core part of the thesis, resulted in the Doubly Trained Surrogate CMA-ES (DTS-CMA-ES). This algorithm uses the uncertainty prediction of a Gaussian process for selecting only a part of the CMA-ES population for evaluation with the expensive objective function while the mean prediction is used for the rest. The DTS-CMA-ES improves upon the state-of-the-art surrogate continuous optimizers in several benchmark tests.

The second part of the thesis investigates the relationship between the predictive accuracy of surrogate models and features of the black-box function landscape. It also studied properties of features for landscape analysis in the context of different transformations and ways of selecting the input data. The landscape analysis is performed on a large set of data generated using the noiseless part of the Comparing Continuous Optimisers benchmark function testbed.

ACKNOWLEDGMENTS

My biggest thanks goes to my family. Especially, I would like to thank to my beloved Kamča, for each moment I spent working on the research and later on this thesis instead of being with you. Thank you for your endless support during the entire process of writing. I would also like to thank to our Emilka, who always brings smile to my face. And of course I am very grateful to my parents for the way they raised and supported me to be a good person.

I am deeply grateful to my supervisor Martin Holeňa. Thank you for your time, suggestions, patience, and of course your belief in abilities of your students.

I must also thank to Lukáš Bajer. Your attitude and the effort you gave to our research in my first years of my doctorate was always inspiring for me and made me continue until I finished all the necessary work.

I would also thank to my friend Jirka who supports me in reaching my goals.

CONTENTS

1 INTRODUCTION 1
1.1 Motivation . . . 1
1.2 Main Contributions . . . 2
1.3 Thesis Outline . . . 3
2 BACKGROUND 5
2.1 Black-Box Optimization . . . 5
2.2 Evolutionary Computation . . . 7
2.2.1 Evolutionary Algorithms . . . 7
2.2.2 CMA-ES . . . 9
2.3 Surrogate Modeling . . . 12
2.3.1 Regression Methods for Optimization . . . 13
2.3.2 Evolution Control . . . 23
2.3.3 Surrogate Versions of CMA-ES . . . 24
2.4 Other Approaches to Black-Box Optimization . . . 29
2.5 Exploratory Analysis of Fitness Landscapes . . . 32
2.6 Optimization Benchmarking . . . 34
2.6.1 COCO framework . . . 35
2.6.2 Postprocessing of Convergence Plot . . . 36
3 GOALS 39
4 CONTRIBUTIONS TO EVOLUTION CONTROL OF SURROGATE MODELS 41
4.1 Model Training in Surrogate-assisted Optimization . . . 41
4.1.1 Training Sets . . . 42
4.1.2 Surrogate Model Training . . . 42
4.2 Surrogate CMA-ES . . . 43
4.2.1 Benchmarking S-CMA-ES using Gaussian Processes and Random Forests 43
4.2.2 Comparison of IPOP and BIPOP versions of S-CMA-ES . . . 45
4.2.3 Conclusion . . . 51
4.3 Adaptive Surrogate-CMA-ES . . . 55
4.3.1 Adaptive Evolution Control for Surrogate CMA-ES . . . 55
4.3.2 Comparison of Adaptive and Static Versions of S-CMA-ES . . . 58
4.3.3 Conclusion . . . 61
4.4 Doubly Trained S-CMA-ES . . . 64
4.4.1 Self-adaptation of the original-evaluated points ratio α . . . 67
4.4.2 Configuration of the Gaussian Process Models . . . 76
4.4.3 Evaluation of Gaussian Process Based CMA-ES Algorithms . . . 81
4.4.4 Conclusion . . . 87
4.5 Comparison of Ordinal and Metric Gaussian Process Regression . . . 94
4.5.1 Implementation of Ordinal GP . . . 94
4.5.2 Experiments on the COCO platform . . . 98
4.5.3 Conclusion . . . 101
4.6 Automated Selection of Covariance Function for GP Surrogate Models . . . 103
4.6.1 Model Selection . . . 104

4.6.2 Experimental Evaluation . . . 106
4.6.3 Conclusion . . . 110
4.7 Boosted Regression Forest for the DTS-CMA-ES . . . 110
4.7.1 Evaluation of Boosted Regression Forest for the DTS-CMA-ES . . . 110
4.7.2 Conclusion . . . 112
4.8 Interaction between Model and Its EC in Surrogate-Assisted CMA-ES . . . 112
4.8.1 Analysis of the EC in Important Surrogate-Assisted CMA-ES Variants . . . 116
4.8.2 Experimental Investigation of Interactions between Model and Its Control . . . 120
4.8.3 Conclusion . . . 126
4.9 Combining Gaussian Processes with Neural Networks for the CMA-ES . . . 126
4.9.1 Using Past Experience for GP Configuration . . . 127
4.9.2 Conclusion . . . 131
5 CONTRIBUTIONS TO LANDSCAPE ANALYSIS OF SURROGATE MODELS 135
5.1 Relations of Surrogate Models to Fitness Landscapes in Expensive Optimization . . . 135
5.1.1 Experimental Investigation of Landscape Analysis for RF and GP . . . 136
5.1.2 Conclusion . . . 140
5.2 Landscape Features Investigation . . . 140
5.2.1 Problem Statement . . . 140
5.2.2 CMA-ES Landscape Features . . . 141
5.2.3 Experimental Investigation of Landscape Features . . . 142
5.2.4 Conclusion . . . 153
5.3 Landscape Analysis for Surrogate Models in the Evolutionary Black-Box Context . . . 153
5.3.1 Relationships of Landscape Features and Surrogate Models . . . 153
5.3.2 Conclusion . . . 160
6 CONCLUSIONS 167
6.1 Summary of Contributions . . . 167
6.2 Future Directions . . . 168
A LANDSCAPE FEATURES DEFINITIONS 185
A.1 Basic Features Φ_{Basic} . . . 185
A.2 CM Angle Features $\Phi_{\text{CM-Angl}}$ . . . 185
A.3 CM Convexity Features $\Phi_{\text{CM-Conv}}$ . . . 186
A.4 CM Gradient Homogeneity features $\Phi_{\text{CM-Grad}}$ . . . 187
A.5 CMA Features Φ_{CMA} . . . 187
A.6 Dispersion Features Φ_{Dis} . . . 188
A.7 Information Content Features Φ_{Inf} . . . 188
A.8 Levelset Features Φ_{Lvl} . . . 189
A.9 Metamodel Features Φ_{MM} . . . 189
A.10 Nearest Better Clustering Features Φ_{NBC} . . . 189
A.11 PCA Features Φ_{PCA} . . . 190
A.12 y-Distribution Features $\Phi_{\text{y-D}}$ . . . 190

LIST OF ACRONYMS

AIC	Akaike information criterion
ANN	artificial neural network
ARD	automatic relevance determination
aS-CMA-ES	Adaptive Surrogate CMA-ES
BBOB	Black-Box Optimization Benchmarking
BIC	Bayes information criterion
BIPOP-CMA-ES	CMA-ES using two different restart strategies
CART	Classification and Regression Tree
CM	cell-mapping
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
COCO	COmparing Continuous Optimizers
DIC	deviance information criterion
DTS-CMA-ES	Doubly Trained Surrogate CMA-ES
EA	Evolutionary Algorithm
EC	evolution control
ECDF	empirical cumulative distribution function
EGO	Efficient Global Optimization
EGO-CMA	combination of EGO and CMA-ES algorithms
EI	Expected Improvement
ELA	Exploratory Landscape Analysis
EP	Evolutionary Programming
ERDE	expected ranking difference error
ERT	expected running time
ES	Evolution Strategy
FE	function evaluation
GA	Genetic Algorithm

GECCO	Genetic and Evolutionary Computation Conference
GP	Gaussian process
GeP	Genetic Programming
GPOP	Gaussian Process Optimization Procedure
IPOP-CMA-ES	CMA-ES using doubling restart strategy
lmm-CMA	local meta-model CMA-ES
lq-CMA-ES	linear-quadratic Global Surrogate Assisted CMA-ES
LS-CMA-ES	Least-Square CMA-ES
LWR	locally weighted regression
MA-ES	Metamodel-Assisted Evolution Strategy
MLE	maximum likelihood estimation
MSE	mean-squared error
NBC	nearest better clustering
OC1	decision tree algorithm using hill-climbing splitting hyperplane
PAIR	decision tree splitting method based on pairs of points
PLSOR	probabilistic least squares ordinal regression
PoI	Probability of Improvement
RBF	radial basis function
RDE	ranking difference error
RF	random forest
RMSE	root mean-squared error
^{s*}ACM-ES	self-adaptive surrogate-assisted CMA-ES
S-CMA-ES	Surrogate CMA-ES
SAPEO	Surrogate-Assisted Partial Order-Based Evolutionary Optimization Algorithm
SECRET	scalable linear regression tree algorithm
SGA	Simple Genetic Algorithm
SMAC	Sequential Model-based Algorithm Configuration
STEP	select the easiest point
SUPPORT	residual splitting method for decision tree

SVR	support vector regression
TSS	training set selection
WAIC	Widely applicable information criterion
ZOE	zero-one error

1 INTRODUCTION

1.1 MOTIVATION

Optimization of an expensive objective or fitness function plays an important role in many engineering and research tasks. For such functions, it is sometimes difficult to find an exact analytical formula. Instead, values for a given input are possible to be obtained only through expensive and time-consuming measurements or experiments. Such functions can be found, for example, in engineering design optimization (Bekasiewicz and Koziel, 2017), computational fluid dynamics simulations (Lee et al., 2016), or in the search for optimal materials (Baerns and Holeňa, 2009; Zaefferer et al., 2016). Those functions are called black-box, and because of their evaluation costs, the primary criterion for assessment of the black-box optimizers is the number of fitness function evaluations necessary to achieve the optimal value.

In order to decrease the number of evaluations of the costly black-box function and still produce reasonably good solutions, a surrogate model of the black-box function can be employed (Ong et al., 2003). Such models are built using the previous evaluations of the black-box function, and then are used to predict the values of new points instead of the original function.

Nowadays, the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) by Hansen (2006) is one of the most robust algorithms for real-world problems and is considered to be the state-of-the-art of continuous black-box optimization. The CMA-ES profits from its invariance properties, particularly on ill-conditioned or highly multi-modal functions, or in high-dimensional spaces when the allowable number of function evaluations per dimension is higher than roughly several hundreds.

The CMA-ES learns some core characteristics of the fitness via perturbations of its covariance matrix and the step-size. Nevertheless, the exact information from the passed evaluations can be used more intensively via surrogate modeling. While spending extra computation time for model construction, surrogate models have been shown to save some of the evaluations of the original black-box function; see the research by Jin (2011) or Rios and Sahinidis (2012). Surrogate models have been increasingly often used in many optimization algorithms, including the CMA-ES. The acceleration of the CMA-ES via surrogates was shown, for example, by Kern et al. (2006), Loshchilov et al. (2013b) or Hansen (2019) and it also constitutes one of research objectives of this thesis.

As can be expected, different surrogate models can show differences in the performance of the same optimizer on different data. Such performance can also change during the algorithm run due to varying landscape of the black-box function in different parts of the search space. Moreover, the predictive accuracy of individual models is strongly influenced by the choice of their parameters. Therefore, investigation of the relationships between the settings of individual models and the optimized black- box objective function is necessary for better understanding of the whole surrogate modeling task.

In last few years, research into landscape analysis of objective functions (cf. the overview by Kerschke (2017b)) has emerged in the context of algorithm selection and algorithm configuration. However, to our knowledge, such features have been investigated in connection with surrogate models in the context of black-box optimization very rarely (Saini et al., 2019; Yu

et al., 2016) and the analysis of landscape features themselves also received small attention so far and not in the context of surrogate models (Renau et al., 2019, 2020). Therefore, these two aspects of landcape analysis in the context of surrogate modeling constitutes the remaining research objectives of this thesis.

1.2 MAIN CONTRIBUTIONS

The first part of the presented contributions uses coupling surrogate models with the CMA-ES, in order to address specific issues of black-box optimization, while preserving all the invariance and robustness properties of the CMA-ES.

The second part investigates features representing the fitness landscapes, their properties and their connection to surrogate models in different scenarios.

Evolution Control of Surrogate Models

As a new contribution to the development of surrogate models for the CMA-ES, we introduced two novel algorithms: *Surrogate CMA-ES* (S-CMA-ES) and *Doubly Trained Surrogate CMA-ES* (DTS-CMA-ES). The former switches between the CMA-ES iterations evaluating the population with the original fitness and the iterations using two kinds of surrogate model. The latter uses a surrogate model for the most promising points in every iteration, and the selection of these points relies on the model's ability to estimate the whole distribution of the predicted fitness value.

S-CMA-ES We have employed the generation-based evolution control by Jin et al. (2001) to evaluate points sampled by the CMA-ES. In such evolution control, the entire population of one generation is evaluated using the original fitness or by the surrogate model built using the original-evaluated data if the model has enough training points. Otherwise, the next generations are sampled and evaluated with the original fitness until the number of training points is sufficient. The model employs Gaussian processes (Rasmussen and Williams, 2006) or random forest (Breiman, 2001), both described in Section 2.3.1. Furthermore, we have presented an adaptive improvement based on a general procedure introduced with the s*ACM-ES algorithm by Loshchilov et al. (2013b), in which the number of generations using the surrogate model before retraining is adjusted depending on the performance of the last instance of the surrogate.

DTS-CMA-ES We have developed the S-CMA-ES successor replacing the generation evolution control by the *doubly trained evolution control*, which utilizes the ability of some models (e. g., Gaussian processes or random forest) to provide the distribution of predicted points. The most promising points according to various kinds of uncertainty criteria are then evaluated using the original black-box fitness. Similarly to S-CMA-ES, we have also investigated an extension of DTS-CMA-ES which controls the usage of the model according to the model's error. The employed models cover different variants of Gaussian processes (Rasmussen and Williams, 2006; Srijith et al., 2012a; Calandra et al., 2016), random forests (Breiman, 2001; Chen and Guestrin, 2016), and also two kinds of polynomial models (Kern et al., 2006; Hansen, 2019).

Landscape analysis of Surrogate Models

One of findings from the contributions to evolution control of surrogate models stated that no surrogate model or algorithm using the model improved the CMA-ES significantly better than all other proposed surrogate model approaches in the benchmark expensive scenario. Therefore, we studied the following aspects of features describing the fitness landcape in connection with the surrogate modeling:

- ◇ connection of landcape features with surrogate models in the context of black-box optimization using static settings only, where the model is selected once at the beginning of the optimization process similarly to Saini et al. (2019),
- ◇ properties of features representing the fitness landscape in the context of the data from actual runs of a surrogate-assisted version of the CMA-ES,
- ◇ connection of landscape features with CMA-ES assisted by surrogate models based on GPs, RFs, and polynomials using different settings of their parameters and the criteria for selecting points for their training in the same context as the previous point.

Contributions in the two latter points are especially crucial due to completely different properties of data from runs of a surrogate-assisted algorithm compared with generally utilized sampling strategies as used in the first point, which imply different values of landscape features as emphasized in Renau et al. (2020).

1.3 THESIS OUTLINE

Chapter 2 introduces Evolutionary Algorithms in the context of continuous black-box optimization, focusing on the state-of-the-art algorithm CMA-ES and its surrogate-assisted versions. Attention is also paid to important surrogate models. The rest of Chapter 2 is devoted to fitness landscape analysis.

Chapter 3 states our goals in the research into the surrogate-assisted continuous black-box evolutionary optimization.

Chapter 4 presents our contributions to the field of surrogate-assisted continuous black-box optimization focused on the improving of the CMA-ES using particularly Gaussian processes and random forests.

Chapter 5 extends research into surrogate modelling in evolutionary optimization to the analysis of relationships between optimized landscapes and surrogate model prediction errors.

Chapter 6 concludes the thesis by summarizing our contributions and future directions.

2 BACKGROUND

A principal challenge in many research and engineering tasks is the optimization of problems with no information about the mathematical description of the objective function. The objective functions of such tasks are called *black-box functions*. For them, we are able to obtain only values in specified points of the input space. Black-box optimization problems regularly appear in applications where the values of the objective function can be obtained only empirically through measurements, experiments, or via computer simulations.

If the evaluation of the objective function is expensive, evolutionary optimization becomes less useful due to a large amount of evaluations necessary to achieve the optimal value. 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 (Auger et al., 2004; Ong et al., 2005; Ratle, 1998; Ulmer et al., 2003) (cf. also the survey paper by Jin (2011)).

Evolutionary Algorithms and its subfields have become the most successful methods for the optimization of empirical objective functions, especially, the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) proposed by Hansen and Ostermeier (1996) is considered as one of the most successful algorithms in the field of black-box optimization. It will be described in some detail in Section 2.2.2. The investigation of combining surrogate models with the CMA-ES resulted in various algorithms, several of which we will describe in more detail in Section 2.3.3.

The large number of various algorithms and surrogate models suggests that choosing the best-performing optimizer(s) out of them is a difficult and complex task. In last eleven years, features characterizing more or less accurately fitness landscapes have been proposed (cf. the overview by Kerschke (2017b)) to tackle tasks where the algorithm or algorithm settings with the highest performance for an unknown optimization problem is selected, a. k. a. *Algorithm Selection* or *Algorithm Configuration* problem, proposed by Rice (1976) (see Section 2.5).

In Section 2.1, we briefly introduce continuous black-box optimization. Section 2.2 gives more details about evolutionary algorithms, especially, about the state-of-the-art CMA-ES. In Section 2.3, two surrogate models and several algorithms using evolution control in combination with the CMA-ES are presented. Non-evolutionary approaches to continuous black-box optimization are mentioned in Section 2.4. In Section 2.5, we introduce the landscape analysis and discuss its applications. Finally, Section 2.6 gives a brief introduction into comparing algorithms in black-box optimization benchmarking.

2.1 BLACK-BOX OPTIMIZATION

Black-box optimization problems frequently appear in the real world, where the values of a *fitness* or an *objective* function $\mathfrak{b}\mathfrak{b}(\mathbf{x})$ can be obtained only empirically through experiments or via computer simulations, but no mathematical expression is known, neither an explicit one as a composition of known mathematical functions, nor an implicit one as a solution of an explicit equation (e. g., a differential equation). We refer to such a function as *black-box* (see Figure 1). Such an empirical evaluation is sometimes very time-consuming or expensive

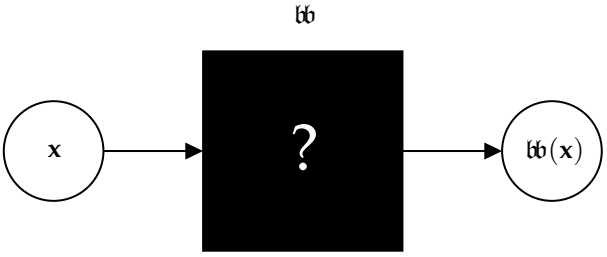


Figure 1: Evaluation of the black box fitness bb in a point **x**.

(Baerns and Holeňa, 2009; Forrester et al., 2008): for example, in the applications of Evolutionary Algorithms to the optimization of chemical materials reported by Holeňa et al. (2012), the evaluation of one generation takes several days to weeks and costs several to many thousands of euros.

The goal of black-box optimization is to find a feasible solution $\mathbf{x}_{\text{opt}}$ of some black-box function $bb : \mathcal{S} \mapsto \mathbb{R}$ with the best value of $bb(\mathbf{x}_{\text{opt}})$, where $\mathcal{S} \subset \mathbb{R}^D$. In case of minimization we can write:

$$\mathbf{x}_{\text{opt}} = \text{argmin}_{\mathbf{x}\in\mathcal{S}}\, bb(\mathbf{x})\,. \tag{1}$$

Since finding the true optimum is often unnecessary or practically impossible in the black-box optimization context, one is often interested in an approximate solution $\hat{\mathbf{x}}_{\text{opt}} \in \mathcal{S}$ called the *optimum of* bb *on* $\mathcal{S}$ *up to a precision* Δbb^* which satisfies

$$\hat{y}^* = bb(\hat{\mathbf{x}}_{\text{opt}}) \in [bb_{\text{opt}},\ bb_{\text{opt}} + \Delta bb^*]\,. \tag{2}$$

Considering that the value bb_{opt} is usually unknown in black-box optimization, it is not convenient to utilize the minimal achieved bb-value as a stopping criterion. On the contrary, in case of benchmarks the bb_{opt} is very well-known.

Because of the evaluation costs, the most important criterion for assessing optimization methods when optimizing an expensive function is the number of fitness function evaluations necessary to reach a reasonable fitness value (i. e., close enough to the optimal value). Thus, we assume that evaluating the fitness represents a considerably higher cost than training a regression model.

The absence of derivatives or explicit fitness formula of the optimized function excludes the application of traditional optimization algorithms like successors of Newton's method (Galántai, 2000) without using any numerical estimates[1]. On the other hand, stochastic methods such as Evolutionary Algorithms are considered to be the most convenient tool for global black-box optimization. Evolutionary Algorithms are described in the following section. Special attention is devoted to the state-of-the-art algorithm CMA-ES. However, Evolutionary Algorithms are not the only useful method for black-box optimization. Other approaches are described in Section 2.4.

1 Methods relying on Newton's method require direct calculations of the derivatives. The approximation of derivatives would decrease the convergence of such methods.

For black-box functions, the necessity to replace derivatives through their numerical estimates turns the application of traditional methods of smooth optimization less suitable than in the case of functions for which derivatives can be obtained explicitely. This lead to using of *derivative-free* methods such as Nelder-Mead algorithm (Nelder and Mead, 1965) and Brent's method (Brent, 1973), both described in Section 2.4, and to stochastic optimization methods such as *Evolutionary algorithms* applying the Darwin's theory of natural selection (Darwin, 1859). The field of evolutionary computation has developed to various subfields such as *Evolution Strategies* (Rechenberg, 1973; Schwefel, 1965), *Evolutionary programming* (Fogel et al., 1966), *Genetic Algorithms* (Holland, 1975), and *Genetic Programming* (Cramer, 1985).

In this section, we give a brief overview of Evolutionary Algorithms and its subfields. A special attention is paid to the state-of-the-art algorithm for the single-objective continuous black-box optimization: the Covariance Matrix Adaptation Evolution Strategy.

2.2.1 *Evolutionary Algorithms*

The first proposals of natural selection can be found in ancient Greece. The most widely accepted and contemporary version of this idea was formulated by the biologist and philosopher Charles Robert Darwin. In his book *On the Origin of Species by Means of Natural Selection* (Darwin, 1859), proposes that the individuals of the same species are different from each other. The differences are subsequently inherited by the offspring of these individuals. The survival of strong individuals in contrast with weak individuals in the nature while preserving the size of population is enabled through the redundancy of descendants. Afterwards, successful individuals are better adapted to the environment.

Any type of Evolutionary Algorithm (EA) apply the following principle (Eiben and Smith, 2003): A *population* of *individuals* (candidate solutions of the task) evaluated by a fitness function representing behavior of individuals in a problem-defined environment uses evolutionary mechanisms and operators to create a new population, i. e., the next *generation*. A *selection* mechanism promotes individuals with greater fitness values over the worse ones to be processed by genetic operators. A *crossover* operator combining two (or more) individuals (*parents*) to create the *offspring* and a *mutation* operator modifying individuals by random both generate a new generation.

During the 20th century, the Evolutionary Computation has developed in a rich variety of different approaches. Especially, the expansion of personal computers has led to a significant speed-up of EA research. Therefore, we list only the most widely used approaches and give their brief introduction.

Evolutionary Programming

The idea of Evolutionary Programming (EP) is based on Lawrence Fogel's approach of using the simulated evolution to generate artificial intelligence (Fogel et al., 1966). Fogel utilized evolution to improve finite-state machines predicting the series of finite-alphabet symbols generated through Markov processes and non-stationary time series. This type of prediction requires the ability to predict new states of the environment, which is considered to be the base of some object's intellingence (organism, machine), and to exploit the prediction to return an appropriate symbol as a result.

In EP, only mutation operators are employed to perform the whole search in the input space (Eiben and Smith, 2003). Since finite-state machines are complex structures, mutation operators apply modifications of various machine components. For example, the modification of the output symbol dependency on an inner state, or inner state addition.

Genetic Algorithms

Genetic Algorithms (GAs) proposed by Holland (1975) in *Adaptation in Natural and Artificial Systems* are probably the most common approach of EAs. Typically, the individuals are represented by strings over a finite alphabet, e. g., binary in Holland's Simple Genetic Algorithm (SGA). The most widely used selection operators in GA are *roulette selection*, where the probability of choosing an individual with fitness $\mathfrak{bb}_i$ from population of size n in $\frac{\mathfrak{bb}_i}{\sum_i^n \mathfrak{bb}_i}$, and *tournament selection* choosing n-times the candidate solution with the best fitness function value from random population subset of size $k < n$. Crossover and mutation operators differ according to the representation used. For example, the new solutions in SGA are generated using *multi-point crossover* and *bit-flip mutation*.

Genetic Programming

Genetic Programming (GeP) proposed by Cramer (1985) is an evolutionary optimization technique imitating artificial evolution of computer programs that perform well in a given task. GeP can be viewed as a GA with individuals represented by variable-sized structured object, usually trees and variation operators defined for those trees. The trees consist of functions (as nodes) and terminals (as leaves). Terminals are typically input variables and constants. Node functions can be not only arithmetic and algebraic functions, but also operators specified by a given problem, e. g., conditional operators.

Genetic operators in GeP have to be carefully defined to conserve the structure of individuals (trees). In a conventional *subtree crossover* operator, two randomly selected subtrees of two parents are swapped to create two offspring. In a typical *subtree mutation*, a randomly selected subtree is substituted ty a randomly generated tree.

Evolution Strategies

Evolution Strategies (ESs), introduced by Hans-Paul Schwefel (1965) and Ingo Rechenberg (1973), represent individuals by real numbers in D-dimensional space $\mathbf{x} \in \mathbb{R}^D$. The core of ES are multivariate normal mutations of individuals. In each generation, a population of new offspring $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$, $\lambda \in \mathbb{N}$ is generated as follows:

$$\mathbf{x}_k = \mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{C}), \quad k = 1, \ldots, \lambda\,, \tag{3}$$

where $\mathbf{m} \in \mathbb{R}^D$ is the parent (but also a mean of the mutation distribution), $\sigma > 0$ is a mutation step-size and $\mathbf{C} \in \mathbb{R}^{D \times D}$ is a covariance matrix.

There are two popular selection strategies in ES differing according to the new population-size λ and the number of parents μ. In a *comma selection strategy*, (μ, λ), $\mu < \lambda$ best offspring are chosen to replace all the parents. In a *plus selection strategy*, $(\mu + \lambda)$, μ best individuals from μ parents plus λ offspring are chosen to be the next generation parents. Therefore, in $(1 + 1)$-ES, the parent compete with its child created through mutation. The individual with better fitness will survive to the next generation. The (μ, κ, λ)-ES, first applied by Schwefel

(Schwefel, 1993), is a variant of the $(\mu + \lambda)$-ES where the individuals exceeding the age of κ generations are eliminated from the selection procedure.

2.2.2 *CMA-ES*

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) proposed by Hansen and Ostermeier (1996) has become one of the most successful continuous black-box optimization algorithms. The combination of

- ◇ comparing only the ordering of function values in a particular population, not their absolute values,
- ◇ the way how the algorithm step-size σ is adapted, and
- ◇ the way how the covariance matrix $\mathbf{C}$ is adapted to learn the appropriate mutation distribution

makes the CMA-ES invariant to order-preserving (strictly increasing) transformations of the objective function and to general linear transformations (scale, rotation, translation) of the search space. Therefore, the performace of the algorithm is identical for all the possible transformations of the original problem case mentioned above as for the original case (see Hansen and Auger (2014) for details).

The $(\mu/\mu_w, \lambda)$-CMA-ES is outlined in Algorithm 1 in its *weighted active* CMA-ES version by Hansen and Ros (2010). The default CMA-ES parameter values for respective phases are as follows:

- ◇ Selection and recombination

$$\lambda = 4 + \lfloor 3 \ln D \rfloor \,, \qquad \mu = \left\lfloor \frac{\lambda}{2} \right\rfloor \,, \tag{4}$$

$$w_i = \frac{\ln(\mu + \frac{1}{2}) - \ln i}{\sum_{j=1}^{\mu} \ln(\mu + \frac{1}{2}) - \ln j} \qquad i = 1, \ldots, \mu \,, \tag{5}$$

$$\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \,. \tag{6}$$

- ◇ Step-size control

$$c_\sigma = \frac{\mu_w + 2}{D + \mu_w + 3} \,, \qquad d_\sigma = 1 + c_\sigma + 2 \max\left(0, \sqrt{\frac{\mu_w - 1}{D + 1}} - 1\right) \,. \tag{7}$$

- ◇ Covariance matrix adaptation

$$c_c = \frac{4}{D + 4} \,, \qquad c_1 = \frac{2 \min\left(1, \frac{\lambda}{6}\right)}{(D + 1.3)^2 + \mu_w} \,,$$
$$c_\mu = \frac{2(\mu_w - 2 + \frac{1}{\mu_w})}{(D + 2)^2 + \mu_w} \,, \qquad c^- = \frac{\mu_w}{4(D + 2)^{1.5} + 2\mu_w} \,, \tag{8}$$

$$\alpha_{\text{old}}^- = \frac{1}{2} \,. \tag{9}$$

Algorithm 1 The $(\mu/\mu_w, \lambda)$-CMA-ES

Input: population-size λ, original fitness function $\mathfrak{bb}$, number of parents μ, selection and recombination parameters $(w_i)_{i=1}^{\mu}$, μ_w, step-size control parameters c_σ, d_σ, covariance matrix adaptation parameters c_c, c_I, c_μ, c^-, α_{old}^-

1: **initialize** $g = 0, \sigma^{(0)} > 0, \mathbf{m}^{(0)} \in \mathbb{R}^D, \mathbf{C}^{(0)} = \mathbf{I}, \mathbf{p}_\sigma^{(0)} = \mathbf{0}, \mathbf{p}_c^{(0)} = \mathbf{0}$
2: **repeat**
3: $\quad \mathbf{x}_k \leftarrow \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)}) \qquad k = 1, \ldots, \lambda$
4: $\quad \mathfrak{bb}_k \leftarrow \mathfrak{bb}(\mathbf{x}_k) \qquad k = 1, \ldots, \lambda$
5: $\quad \mathbf{m}^{(g+1)} \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
6: $\quad \mathbf{p}_\sigma^{(g+1)} \leftarrow (1-c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)}\sqrt{\mu_w}\left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$
7: $\quad h_\sigma \leftarrow \mathbb{I}\left(\|\mathbf{p}_\sigma^{(g+1)}\| < \sqrt{1-(1-c_\sigma)^{2(g+1)}}\left(1.4 + \frac{2}{D+1}\right) \mathbb{E}\|\mathcal{N}(0,1))\| \right)$
8: $\quad \mathbf{p}_c^{(g+1)} \leftarrow (1-c_c)\mathbf{p}_c^{(g)} + h_\sigma \sqrt{c_c(2-c_c)}\sqrt{\mu_w}\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$
9: $\quad \mathbf{y}_{i:\lambda} \leftarrow \frac{\mathbf{x}_{i:\lambda} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$
10: $\quad \mathbf{C}_\mu^+ \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^\top$
11: $\quad \mathbf{y}_{\lambda-i:\lambda} \leftarrow \frac{\left\| \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \left(\mathbf{x}_{\lambda-\mu+1+i:\lambda} - \mathbf{m}^{(g)}\right) \right\|}{\left\| \left(\mathbf{C}^{(g)}\right)^{-\frac{1}{2}} \left(\mathbf{x}_{\lambda-i:\lambda} - \mathbf{m}^{(g)}\right) \right\|} \times \frac{\mathbf{x}_{\lambda-i:\lambda} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$
12: $\quad \mathbf{C}_\mu^- \leftarrow \sum_{i=0}^{\mu-1} w_{i+1} \mathbf{y}_{\lambda-i:\lambda}\mathbf{y}_{\lambda-i:\lambda}^\top$
13: $\quad \mathbf{C}^{(g+1)} \leftarrow (1 - c_1 - c_\mu + c^-\alpha_{\text{old}}^-)\mathbf{C}^{(g)} + c_1 \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^\top}$
$\qquad\qquad + (c_\mu + c^-(1-\alpha_{\text{old}}))\mathbf{C}_\mu^+ - c^-\mathbf{C}_\mu^-$
14: $\quad \sigma^{(g+1)} \leftarrow \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1 \right) \right)$
15: $\quad g \leftarrow g + 1$
16: **until** stopping criterion is met

Output: $\mathbf{x}_{\text{res}}$ (resulting optimum)

Sampling & Mean Update

After generating λ new candidate solutions using mean $\mathbf{m}^{(g)}$ of the mutation distribution added to a random Gaussian mutation with covariance matrix $\mathbf{C}^{(g)}$ (step 3), the new offspring are evaluated on the fitness function $\mathfrak{bb}$ (step 4). The new mean $\mathbf{m}^{(g+1)}$ of the mutation distribution is computed as the weighted sum of μ best points from the λ ordered offspring $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ (step 5), where the symbol $i : \lambda$ denotes i-th best individual on $\mathfrak{bb}$. The weights of better ranked are usually chosen higher; therefore, the better the rank, the stronger the influence of the individual on the mean.

Evolution Path

The sum of consecutive successful mutation steps of the algorithm in the search space $\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$ is called an *evolution path* and it is denoted $\mathbf{p}_\sigma$ (step 6). Successful steps are tracked in the sampling space using transformation $\mathbf{C}^{(g)^{-\frac{1}{2}}}$, which is symmetric positive definite, and of

course, $\mathbf{C}^{(g)^{-\frac{1}{2}}}\mathbf{C}^{(g)^{-\frac{1}{2}}} = \left(\mathbf{C}^{(g)}\right)^{-1}$. Therefore, $\mathbf{u}_k = \mathbf{C}^{(g)^{-\frac{1}{2}}}\frac{\mathbf{x}_k}{\sigma^{(g)}}$, where $\mathbf{x}_k$ is a sampled point and $\mathbf{u}_k$ is its preimage in a space of sampling. Note that, $\mathbf{C}^{(g)^{-\frac{1}{2}}} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^{-1}}\mathbf{B}^{(g)^\top}$, where $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^2}\mathbf{B}^{(g)^\top}$ is an eigendecomposition of $\mathbf{C}^{(g)}$, where $\mathbf{B}^{(g)}$ is an orthonormal basis of eigenvectors, and $\mathbf{D}^{(g)}$ is a diagonal matrix of square roots of eigenvalues. The two remaining evolution path elements are a decay factor c_σ decreasing the impact of successful steps with increasing generations, and μ_w used to normalize the variance of $\mathbf{p}_\sigma$.

Covariance Matrix Adaptation

Two parts of the adaptation of covariance matrix are *rank-μ* update and *rank-one* update (step 13). The rank-one update computes evolution path $\mathbf{p}_c$ in the similar way $\mathbf{p}_\sigma$ is computed (in step 6); however, the coordinate system is not changed (step 8). If σ increases too quickly, h_σ indicator is used to stop the update.

The rank-μ update cummulates successful steps of μ best individuals in matrix $\mathbf{C}_\mu^+$ (step 10). The weighted active CMA-ES by Hansen and Ros (2010) utilizes μ worst individuals to decrease the variance of the mutation distribution in unpromising directions (Jastrebski and Arnold, 2006) by computing $\mathbf{C}_\mu^-$ (step 12).

Step-size Control

The step-size σ (step 14) is updated according to the ratio between the evolution path length $\|\mathbf{p}_\sigma\|$ the expected length of a random evolution path. If the ratio is greater than 1, the step-size is increasing, and decreasing otherwise.

Restart Strategies

The CMA-ES uses restart strategies to deal with multimodal fitness landscapes and to avoid being trapped in local optima. A multi-start strategy where the population size is doubled in each restart presented by Auger and Hansen (2005) is referred to as IPOP-CMA-ES.

The BIPOP-CMA-ES by Hansen (2009a), unlike IPOP-CMA-ES, considers two different restart strategies. In the first one, corresponding to the IPOP-CMA-ES, the population size is doubled in each restart i_{restart} using a constant initial step-size $\sigma^0_{\text{large}} = \sigma^0_{\text{default}}$:

$$\lambda_{\text{large}} = 2^{i_{\text{restart}}}\lambda_{\text{default}}\,. \tag{10}$$

In the second one, the smaller population size λ_{small} is computed as

$$\lambda_{\text{small}} = \left\lfloor \lambda_{\text{default}}\left(\frac{1}{2}\frac{\lambda_{\text{large}}}{\lambda_{\text{default}}}\right)^{\mathcal{U}[0,1]^2}\right\rfloor, \tag{11}$$

where $\mathcal{U}[0,1]$ denotes the uniform distribution in $[0,1]$. The initial step-size is also randomly drawn as

$$\sigma^0_{\text{small}} = \sigma^0_{\text{default}} \times 10^{-2\mathcal{U}[0,1]}\,. \tag{12}$$

The BIPOP-CMA-ES performs the first run using the default population size λ_{default} and the initial step-size $\sigma^0_{\text{default}}$. In the following restarts, the strategy with fewer function evaluations summed over all algorithm runs is selected.

As mentioned earlier, Evolutionary Algorithms typically need many fitness evaluations to reach a given target. Surrogate modeling, originally from the field of smooth optimization, is an approach to reduce the number of expensive objective function evaluations through using its regression model (Ong et al., 2003). This model, a. k. a. *surrogate model*, is trained on the already available input–output value pairs $(\mathbf{x}_i, y_i), i = 1, \ldots, N$ and is used instead of the original expensive fitness to evaluate some of the points needed by the optimization algorithm. To our knowledge, the following types of regression models were employed so far in single-objective continuous black-box optimization:

- low degree polynomials (Auger et al., 2004; Hansen, 2019; Kern et al., 2006; Liang et al., 2000), which are models in the spirit of traditional *response surface models* by Myers and Montgomery (1995);
- *artificial neural networks* (ANNs), in particular multilayer perceptrons and *radial basis function* (RBF) networks (Bajer and Holeňa, 2010; Jin et al., 2005; Sun et al., 2017; Zhou et al., 2007);
- *support vector regression* (SVR), either metric or ordinal (Loshchilov et al., 2012, 2013b; Ulmer et al., 2004);
- *Gaussian processes* (GPs), a. k. a. kriging (Büche et al., 2005; Emmerich et al., 2006; Kruisselbrink et al., 2010; Ulmer et al., 2003).

The investigation of combining surrogate models with the CMA-ES resulted in various algorithms, several of which we will describe in the following text.

Jin et al. (2001) proposed the following two evolution control strategies for the utilization of surrogate models in the CMA-ES: *individual-based* strategy evaluates λ points using the original fitness function selected out of a larger set of points evaluated by the surrogate model. In *generation-based* strategy, the entire population is evaluated on the original fitness function for η generations and on the model for the subsequent $\kappa > \eta$ generations. The proposed surrogate model in this paper were ANNs, as well as, a year later in a next paper also by Jin et al. (2002).

The individual evolution control was employed by Emmerich et al. (2002) proposing the first GP based selection strategy for the CMA-ES in *Metamodel-Assisted Evolution Strategy* (MA-ES).

Auger et al. (2004) have used Least-Square minimization in LS-CMA-ES to train a quadratic model of the fitness function for covariance matrix adaptation in the CMA-ES. This surrogate model employed automatic detection of the model inaccuracy allowing to switch between the original CMA-ES and the surrogate model.

One year later, *Gaussian Process Optimization Procedure* (GPOP) by Büche et al. (2005) suggests a different approach to surrogate modeling: in each GPOP generation, a local GP model is constructed around the so-far-best solution and then the model is directly optimized by the CMA-ES to find its optimum which is subsequently evaluated using the original fitness function.

An effective combination of building local surrogate models and controlling changes in population ranking after fitness function evaluation is incorporated in the *local meta-model CMA-ES* (lmm-CMA) proposed by Kern et al. (2006) and later improved by Auger et al. (2013).

In Kruisselbrink et al. (2010), GP was combined with the CMA-ES in order to find robust solutions on noisy functions in *Kriging metamodeling based CMA-ES*. The algorithm builds a local GP model for each offspring using the points from an archive evaluated with the original noisy fitness function and subsequently estimates the function values without noise. The original fitness evaluation is performed only if the archive does not contain a representative sample set for the surrogate model construction. The covariance matrix of the CMA-ES is also employed to transform the input space.

A different usage of regression models than the combination with the CMA-ES presents the *Sequential Model-based Algorithm Configuration* (SMAC) by Hutter et al. (2011). The surrogate models are built and used in a parameter space of algorithm settings.

The ^{s*}ACM-ES proposed by Loshchilov et al. (2012) employs ordinal SVR to estimate the ordering of the fitness function values.

Another surrogate-assisted approach using an ensemble of local GP models sharing the same parameters has been proposed by Lu et al. (2013). The algorithm selects the best points out of a larger population evaluated using the model according to one of several implemented strategies.

In Mohammadi et al. (2015), the combination of the *Efficient Global Optimization* (EGO) algorithm by Jones et al. (1998) using GP for direct optimization and the CMA-ES resulted in the EGO-CMA algorithm.

Combination of the GP predicted mean and variance using dominance relations to order a CMA-ES population was employed in the *Surrogate-Assisted Partial Order-Based Evolutionary Optimization Algorithm* (SAPEO) by Volz et al. (2017).

Most recently, Hansen (2019) presented his *linear-quadratic Global Surrogate Assisted CMA-ES* (lq-CMA-ES), using an at most quadratic polynomial to enhance CMA-ES performance.

This section presents two main strategies how to employ a surrogate model in evolutionary algorithms (Jin, 2005). We also give some details to two regression methods for surrogate modeling. In addition, we describe several algorithms utilizing surrogate models in connection with the CMA-ES. At the end of this section, we briefly present few methods for the selection of training points for the surrogate model.

2.3.1 *Regression Methods for Optimization*

In this section, we list two regression methods for surrogate modeling in evolutionary optimization: Gaussian processes and random forests. These two model differ from other common surrogate models through estimating the whole probability distribution of fitness values.

To combine Gaussian processes 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. In addition, research into relationships of asymptotic properties of important kinds of ANN to properties of GP (Lee et al., 2018; Matthews et al., 2018; Novak et al., 2019) have incited attempts to integrate them together (Bui et al., 2016; Calandra et al., 2016; Cutajar et al., 2017; Hebbal et al., 2018; Hernández-Muñoz et al., 2020; Wilson et al., 2016). Therefore, we include also a description of the ordinal variant of GPs and a description of GPs integrating them into an ANN.

Gaussian Processes

Gaussian process on a D-dimensional Euclidean space $\mathcal{X} \subseteq \mathbb{R}^D$ is a probabilistic model based on Gaussian distributions. It is a collection of random variables $\mathrm{GP}_{\mathcal{X}} = (f(\mathbf{x}))_{\mathbf{x}\in\mathcal{X}}$, indexed by the space $\mathcal{X}$, such that each finite subcollection has a joint multi-dimensional normal, i. e., Gaussian distribution (Rasmussen and Williams, 2006).

The GP is completely specified by a *mean function* $\mu(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$, typically assumed constant, and by a *covariance function* $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ such that for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$,

$$\mathbb{E}f(\mathbf{x}) = \mu, \;\; \mathrm{cov}(f(\mathbf{x}), f(\mathbf{x}')) = \kappa(\mathbf{x}, \mathbf{x}')\,. \tag{13}$$

Therefore, a Gaussian process is usually denoted $\mathcal{GP}(\mu, \kappa)$ or $\mathcal{GP}(\mu, \kappa(\mathbf{x}, \mathbf{x}'))$.

The value of $f(\mathbf{x})$ is typically accessible only as a noisy observation $y = f(\mathbf{x}) + \varepsilon$, where ε is a zero-mean Gausssian noise with a variance $\sigma_n^2 > 0$. Then

$$\kappa(y, y') = \kappa(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \mathbb{I}(\mathbf{x} = \mathbf{x}')\,, \tag{14}$$

where $\mathbb{I}(p)$ equals 1 when a proposition p is true and 0 otherwise.

PREDICTIONS Using the Gaussian process for prediction always starts with a training set of N points in the input space $\mathcal{X}$,

$$\mathbf{X}_N = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^N$$

for which the function values $\mathbf{y}_N = \{y_i = f(\mathbf{x}_i)\}_{i=1}^N$ are known.

Let us denote the probability density of the multivariate N-dimensional Gaussian distribution of $\mathbf{y}_N$, where $\mathbf{y}_N$ corresponds to $\mathbf{X}_N$, as $p(\mathbf{y}_N \mid \mathbf{X}_N)$[2]. If the mean function μ of the GP is zero, then $p(\mathbf{y}_N \mid \mathbf{X}_N) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_N)$. For prediction, we consider a new, $(N+1)$-st point $(\mathbf{x}^*, y^*)$. The density of the extended vector $\mathbf{y}_{N+1} = (y_1, \ldots, y_N, y^*)$ is (Büche et al., 2005)

$$p(\mathbf{y}_{N+1} \mid \mathbf{X}_{N+1}) = \frac{\exp(-\frac{1}{2}\mathbf{y}_{N+1}^\top \mathbf{C}_{N+1}^{-1} \mathbf{y}_{N+1})}{\sqrt{(2\pi)^{N+1} \det(\mathbf{C}_{N+1})}} \tag{15}$$

where $\mathbf{C}_{N+1}$ is the covariance matrix of the $(N+1)$-dimensional Gaussian distribution. This covariance can be written as

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \kappa(\mathbf{X}_N, \mathbf{x}^*) \\ \kappa(\mathbf{x}^*, \mathbf{X}_N) & \kappa(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix}. \tag{16}$$

$\kappa(\mathbf{X}_N, \mathbf{x}^*) = \kappa(\mathbf{x}^*, \mathbf{X}_N)^\top$ is the vector of covariances between the new point $\mathbf{x}^*$ and the original training data $\mathbf{X}_N$, and $\kappa(\mathbf{x}^*, \mathbf{x}^*)$ is the value of the covariance function for the new point itself. The term $\kappa(\mathbf{x}^*, \mathbf{x}^*)$ is without noise since it is a prediction, not a model of a noisy observation.

In addition, one can use the same template of the covariance matrix (16) for expressing multiple, say t, testing points together. Then, $\kappa(\mathbf{x}^*, \mathbf{X}_N)$ and $\kappa(\mathbf{X}_N, \mathbf{x}^*)$ are replaced with $\kappa(\mathbf{X}^*, \mathbf{X}_N)$ and $\kappa(\mathbf{X}_N, \mathbf{X}^*)$ respectively — the matrices of the covariance function values between a number of testing points and the training set. The scalar $\kappa(\mathbf{x}^*, \mathbf{x}^*)$, analogously, becomes the matrix $\kappa(\mathbf{X}^*, \mathbf{X}^*)$, and the whole covariance matrix is

$$\mathbf{C}_{N+t} = \begin{pmatrix} \kappa(\mathbf{X}_N, \mathbf{X}_N) + \sigma_n^2 \mathbf{I}_N & \kappa(\mathbf{X}_N, \mathbf{X}^*) \\ \kappa(\mathbf{X}^*, \mathbf{X}_N) & \kappa(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}. \tag{17}$$

2 Remember, that N is not the dimension of the input space $\mathcal{X}$, but the number of training points in $\mathbf{X}_N$.

The inverse of the extended covariance matrix $\mathbf{C}_{N+1}^{-1}$ can be obtained using the inverse of the training covariance $\mathbf{C}_N^{-1}$. Further, the vector $\mathbf{y}_N$ of y-values of the training points is known, and so the density of the distribution in a new point reduces to a univariate Gaussian density

$$p(y^* \mid \mathbf{X}_{N+1}, \mathbf{y}_N) \propto \exp\left(-\frac{1}{2}\frac{(y^* - \hat{y}^*)^2}{(\hat{s}^*)^2}\right) \tag{18}$$

with the mean and variance given by

$$\hat{y}^* = \kappa(\mathbf{x}^*, \mathbf{X}_N)\mathbf{C}_N^{-1}\mathbf{y}_N\,, \tag{19}$$

$$(\hat{s}^*)^2 = \kappa(\mathbf{x}^*, \mathbf{x}^*) - \kappa(\mathbf{x}^*, \mathbf{X}_N)\mathbf{C}_N^{-1}\kappa(\mathbf{x}^*, \mathbf{X}_N)^\top\,. \tag{20}$$

Further details can be found in, for example, (Rasmussen and Williams, 2006).

Abandoning the assumption $\mu(\mathbf{x}) = \mathbf{0}$, equation (19) generalizes to

$$\hat{y}^* = \mu(\mathbf{x}^*) + \boldsymbol{\kappa}^\top\, \mathbf{C}_N^{-1}\,(\mathbf{y}_N - \mu(\mathbf{X}_N)) \tag{21}$$

where the mean function is most commonly set to a constant $\mu(\mathbf{x}) = m_\mu$.

Generalizing these equations for multiple testing points, the multivariate posterior distribution of t new points $\mathbf{X}^*$ is

$$\mathbf{y}^* \mid \mathbf{X}_{N+t}, \mathbf{y}_N \sim \mathcal{N}(\hat{\mathbf{y}}^*, \hat{\boldsymbol{\Sigma}}^*), \quad \text{where} \tag{22}$$

$$\hat{\mathbf{y}}^* = \kappa(\mathbf{X}^*, \mathbf{X}_N)\,\mathbf{C}_N^{-1}\,(\mathbf{y}_N - \mu(\mathbf{X}_N)) \tag{23}$$

$$\hat{\boldsymbol{\Sigma}}^* = \kappa(\mathbf{X}^*, \mathbf{X}^*) - \kappa(\mathbf{X}^*, \mathbf{X}_N)\,\mathbf{C}_N^{-1}\,\kappa(\mathbf{X}^*, \mathbf{X}_N)^\top. \tag{24}$$

For the Gaussian process with a vectorized mean function $\mu : \mathcal{X}^t \to \mathbb{R}^t$, the expression (23) changes to

$$\hat{\mathbf{y}}^* = \mu(\mathbf{X}^*) + \kappa(\mathbf{X}^*, \mathbf{X}_N)\,\mathbf{C}_N^{-1}\,(\mathbf{y}_N - \mu(\mathbf{X}_N)).$$

From the computational perspective, the calculation of the covariance matrix $\mathbf{C}_N$ takes $\mathcal{O}(DN^2)$ time, and the complexity of the likelihood calculation for hyperparameter estimation is $\mathcal{O}(N^3)$ due to inversion of $\mathbf{C}_N$. Once the $\mathbf{C}_N^{-1}$ is calculated, the complexity of the prediction is only $\mathcal{O}(N^2)$.

COVARIANCE FUNCTIONS The values of the covariance matrix $\mathbf{C}_N$ are defined by an $N \times N$-dimensional matrix $\mathbf{K}_N$ of values of a covariance function κ and the noise variance σ_n^2 as

$$\mathbf{C}_N = \mathbf{K}_N + \sigma_n^2\mathbf{I}_N\,, \tag{25}$$

where $\{\mathbf{K}_N\}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ being parameters of κ, and $\mathbf{I}_N$ is an identity matrix. A covariance function κ describes prior assumptions on the shape of the modeled function b and expresses the similarity between function values in two points of the input space (Rasmussen and Williams, 2006).

Covariance functions can not be arbitrary, in particular, their values have to form a positive semidefinite matrix $\mathbf{K}_N$. The most commonly used covariances are stationary, i. e., depending only on the distance d of the points $\mathbf{x}$ and $\mathbf{x}'$, between which the covariance is computed. The stationary covariance functions considered in this work include the *squared exponential*

$$\kappa_{\text{SE}}(d; \sigma_f, \ell) = \sigma_f^2 \exp\left(-\frac{d^2}{2\ell^2}\right)\,, \tag{26}$$

which is suitable especially when the modeled function is rather smooth, the *Matérn* covariance function for three values of the parameter $\nu \in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$

$$\kappa_{\text{Mat}}^{1/2}(d; \sigma_f, \ell) = \sigma_f^2 \exp\left(-\frac{d}{\ell}\right), \tag{27}$$

$$\kappa_{\text{Mat}}^{3/2}(d; \sigma_f, \ell) = \sigma_f^2 \left(1 + \frac{\sqrt{3}d}{\ell}\right) \exp\left(-\frac{\sqrt{3}d}{\ell}\right), \tag{28}$$

$$\kappa_{\text{Mat}}^{5/2}(d; \sigma_f, \ell) = \sigma_f^2 \left(1 + \frac{\sqrt{5}d}{\ell} + \frac{5d^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}d}{\ell}\right), \tag{29}$$

and the *rational quadratic*

$$\kappa_{\text{RQ}}(d; \sigma_f, \ell) = \sigma_f^2 \left(1 + \frac{d^2}{2\ell^2 \alpha}\right)^{-\alpha}, \quad \alpha > 0\,. \tag{30}$$

The parameter ℓ is the characteristic length-scale with which the distance of two considered data points is compared and σ_f^2 is the signal variance. Here, we only consider isotropic versions with a scalar characteristic length-scale $\ell > 0$. A generalization through Mahalanobis distance with a non-isotropic distance matrix is called *automatic relevance determination* (ARD).

While squared exponential leads to a smooth process, Matérn functions relax the assumptions of smoothness. The rational quadratic is an approximation of an infinite weighted sum of squared exponentials with weights depending on the summand's characteristic length scale and parameter α (Rasmussen and Williams, 2006).

Stationary functions have a limited capability of capturing global structure (Bengio and Lecun, 2007). A classic example of a non-stationary covariance is a *neural network* covariance derived by placing a Gaussian prior on the weights of a feed-forward neural network (Rasmussen and Williams, 2006)

$$\kappa_{\text{NN}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \arcsin\left(\frac{2\tilde{\mathbf{x}}^\top \mathbf{P} \tilde{\mathbf{x}}'}{\sqrt{(1 + 2\tilde{\mathbf{x}}^\top \mathbf{P} \tilde{\mathbf{x}}')(1 + 2\tilde{\mathbf{x}}^\top \mathbf{P} \tilde{\mathbf{x}}')}}\right), \tag{31}$$

where $\mathbf{P} \in \mathbb{R}^{(D+1)\times(D+1)}$ is the positive definite covariance of the input-to-hidden weights prior and $\tilde{\mathbf{x}} = (1, \mathbf{x}), \tilde{\mathbf{x}}' = (1, \mathbf{x}')$ are inputs augmented by a bias unit. The name of this covariance function is due to the fact that for a simple ANN with D inputs connected to one output, computing the function $f(\mathbf{x}) = \text{err}(w_0 + \sum_{j=1}^{D} w_j \mathbf{x}_j)$ with $\text{err}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \,\mathrm{d}t$, if the weights are random variables and the vector $(w_0, w_1, \ldots, w_D)$ has the distribution $\mathcal{N}(0, \mathbf{P})$, then

$$\mathbb{E} f(\mathbf{x}) f(\mathbf{x}') = \kappa_{\text{NN}}(\mathbf{x}, \mathbf{x}')\,. \tag{32}$$

A dot product with a bias constant term σ_0^2 models *linear* functions:

$$\kappa_{\text{LIN}}(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x}^\top \mathbf{x}'\,. \tag{33}$$

The *quadratic* covariance is the square of a linear covariance:

$$\kappa_{\text{Q}}(\mathbf{x}, \mathbf{x}') = (\sigma_0^2 + \mathbf{x}^\top \mathbf{x}')^2\,. \tag{34}$$

Gibbs (1997) derived a squared exponential covariance function with *spatially varying length-scale*

$$\kappa_{\text{Gibbs}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(\frac{2\ell(\mathbf{x})\ell(\mathbf{x}')}{\ell(\mathbf{x})^2 + \ell(\mathbf{x}')^2} \right)^{\frac{D}{2}} \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^\top (\mathbf{x} - \mathbf{x}')}{\ell(\mathbf{x})^2 + \ell(\mathbf{x}')^2} \right), \tag{35}$$

where $\ell(\mathbf{x})$ is an arbitrary positive function of $\mathbf{x}$. Using variable length-scale can cause behaviour different than one would expect and, therefore, a positive function $\ell(\mathbf{x})$ should be selected carefully (see Gibbs (1997) for further details). In the present work, we used a logarithm of a linear function.

Duvenaud et al. (2011) derived *additive* covariance function through assigning each dimension a one-dimensional base covariance function $\kappa_i(\mathbf{x}_i, \mathbf{x}_i'), i = 1, \ldots, D$

$$\kappa_{\text{ADD}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \prod_{i=1}^{D} \kappa_i(\mathbf{x}_i, \mathbf{x}_i') \,. \tag{36}$$

In the case where each κ_i is a one-dimensional squared-exponential kernel, κ_{ADD} corresponds to the multivariate squared-exponential kernel, a. k. a. Gaussian kernel:

$$\kappa_{\text{ADD}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \prod_{i=1}^{D} \kappa_i(\mathbf{x}_i, \mathbf{x}_i') = \sigma_f^2 \prod_{i=1}^{D} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_i')^2}{2\ell_i^2} \right) = \sigma_f^2 \exp\left(-\sum_{i=1}^{D} \frac{(\mathbf{x}_i - \mathbf{x}_i')^2}{2\ell_i^2} \right). \tag{37}$$

Spectral mixture of Q components, $Q \in \mathbb{N}$ (Wilson and Adams, 2013; Wilson et al., 2016):

$$\kappa_{\text{SM}}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} c_q \sqrt{(2\pi\ell_q^2)^D} \prod_{j=1}^{D} \exp(-2\pi^2 \ell_q^2 (\mathbf{x}_j - \mathbf{x}_j')^2) \cos(2\pi(\boldsymbol{\mu}_q)_j (\mathbf{x}_j - \mathbf{x}_j')) \,, \tag{38}$$

where $c_q \in \mathbb{R}, \ell_q > 0, \boldsymbol{\mu}_q \in \mathbb{R}^D, q = 1, \ldots, Q$. Its name is due to the fact that the function S defined $S(\mathbf{s}) = \sqrt{(2\pi\ell^2)^D} \exp(-2\pi^2 \ell^2 \|\mathbf{s}\|^2)$ is the spectral density of κ_{SE}.

The closer the points $\mathbf{x}$ and $\mathbf{x}'$ are, the closer the covariance function value is to 1 and the stronger correlation between the function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ is. Our choice of the covariance functions was motivated by its simplicity and the possibility of finding the hyper-parameter values by the maximum-likelihood method.

ORDINAL GAUSSIAN PROCESSES 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 (Chu and Ghahramani, 2005; Srijith et al., 2012b,a). In (Srijith et al., 2012a), they have been compared on 9 data sets from the UCI Machine learning repository (University of California, Irvine) and their average predictive performance was similar. Therefore, we will describe the *probabilistic least squares ordinal regression* (PLSOR) by Srijith et al. (2012a) 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, which will be described in detail in Section 4.5.1.

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 $\mathcal{N}(\mu, \sigma^2)$ is mapped to a particular interval $I_k, k = 1, \ldots, r$, is

$$P(f(\mathbf{x}) \in I_k) = \Phi\left(\frac{b_k - (\alpha_0 - \alpha\mu)}{\sqrt{1 + \alpha^2\sigma^2}}\right) - \Phi\left(\frac{b_{k-1} - (\alpha_0 - \alpha\mu)}{\sqrt{1 + \alpha^2\sigma^2}}\right) = \\ = \Phi\left(\frac{\alpha\mu + \beta_k}{\sqrt{1 + \alpha^2\sigma^2}}\right) - \Phi\left(\frac{\alpha\mu + \beta_{k-1}}{\sqrt{1 + \alpha^2\sigma^2}}\right), \tag{39}$$

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 (39), the PLSOR approach estimates the likelihood of a particular realization y_i of the random variable $f(\mathbf{x}_i)$, $i = 1, \ldots, n$ as the probability that the random variable $f(\mathbf{x}_i)$ 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 (19) and (20) but with the covariance function k there using 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 $I_1, \ldots, I_r$:

$$\hat{\mathcal{L}}(y_i \in I_{y_i}, i = 1, \ldots, n | \{\mathbf{x}_l\}_{l=1}^n, \alpha, \beta_1, \ldots, \beta_{r-1}, \boldsymbol{\theta}) = \\ \prod_{i=1}^{n} \left(\Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i}}{\sqrt{1 + \alpha^2\sigma^2_{-i}}}\right) - \Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i - 1}}{\sqrt{1 + \alpha^2\sigma^2_{-i}}}\right) \right). \tag{40}$$

From (40), both the hyperparameters $\boldsymbol{\theta}$ and the relative positions $\beta_1, \ldots, \beta_{r-1}$ of thresholds are simultaneously estimated.

GP AS THE OUTPUT LAYER OF A NEURAL NETWORK The approach integrating GPs into an ANN as its output layer has been independently proposed by Calandra et al. (2016) and by Wilson et al. (2016). It relies on the following two assumptions:

1. If n_I denotes the number of the ANN input neurons, then the ANN computes a *mapping* net *of n_I-dimensional input values into the set $\mathcal{X}$* on which is the GP. Consequently, the number of neurons in the last hidden layer equals the dimension D, and the ANN maps an input v into a point $\mathbf{x} = \text{net}(\mathbf{v}) \in \mathcal{X}$, corresponding to an observation $\mathfrak{bb}(\mathbf{x} + \varepsilon)$ governed by the GP (Figure 2). From the point of view of the ANN inputs, we get $\mathcal{GP}(\mu(\text{net}(\mathbf{v})), \kappa(\text{net}(\mathbf{v}), \text{net}(\mathbf{v}')))$.

2. The GP mean μ *is assumed to be a known constant*, thus not contributing to the GP hyperparameters and independent of net.

Due to the second assumption, the GP depends only on the parameters $\boldsymbol{\theta}^\kappa$ of the covariance function. As to the ANN, it depends on the one hand on the vector $\boldsymbol{\theta}^W$ of its weights and

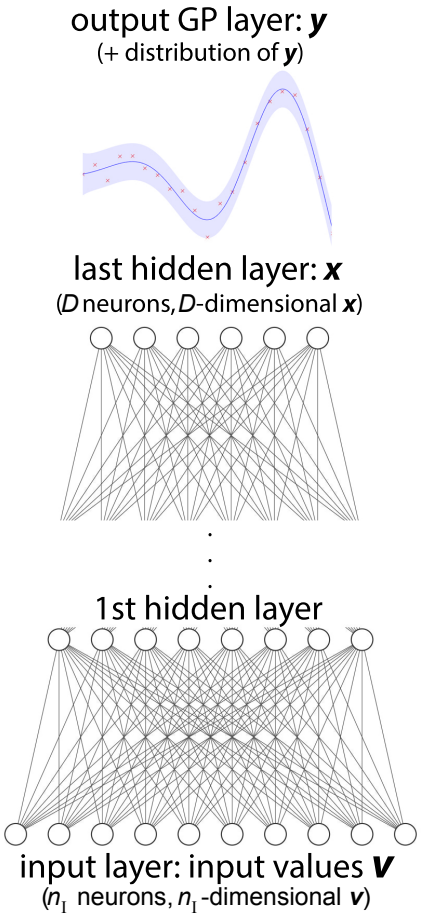


Figure 2: Schema of the integration of a GP into an ANN as its output layer.

biases, on the other hand on the network architecture, which we will treat as fixed before network training starts.

Consider now n inputs to the neural network, $v_1, \ldots, v_n$, mapped to the inputs $x_1 = \mathrm{net}(v_1)$, $\ldots$, $x_n = \mathrm{net}(v_n)$ of the GP, corresponding to the observations $\mathbf{y} = (y_1, \ldots, y_n)^\top$. Then the log-likelihood of $\boldsymbol{\theta}$ is

$$\mathcal{L}(\boldsymbol{\theta}) = \ln p(\mathbf{y}; \mu, \kappa, \sigma_n^2) = \tag{41}$$

$$= -\frac{1}{2}(\mathbf{y} - \mu)^\top \mathbf{K}^{-1}(\mathbf{y} - \mu) - \ln(2\pi) - \frac{1}{2}\ln\det(\mathbf{K}), \tag{42}$$

where μ is the constant assumed in 2., and

$$(\mathbf{K})_{i,j} = \kappa(\mathrm{net}(v_i), \mathrm{net}(v_j))\,. \tag{43}$$

Let model training, searching for the vector $(\boldsymbol{\theta}^\kappa, \boldsymbol{\theta}^W)$, be performed in the simple but, in the context of neural networks, also the most frequent way – as gradient descent. The partial derivatives forming $\nabla_{(\boldsymbol{\theta}^\kappa, \boldsymbol{\theta}^W)}\mathcal{L}$ can be computed as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}^\kappa_\ell} = \sum_{i,j=1}^{n} \frac{\partial \mathcal{L}}{\partial \mathbf{K}_{i,j}} \frac{\partial \mathbf{K}_{i,j}}{\partial \boldsymbol{\theta}^\kappa_\ell}, \tag{44}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}^W_\ell} = \sum_{i,j,k=1}^{n} \frac{\partial \mathcal{L}}{\partial \mathbf{K}_{i,j}} \frac{\partial \mathbf{K}_{i,j}}{\partial \mathbf{x}_k} \frac{\partial \operatorname{net}(v_k)}{\partial \boldsymbol{\theta}^W_\ell}. \tag{45}$$

In (44), the partial derivatives $\frac{\partial \mathcal{L}}{\partial \mathbf{K}_{i,j}}$, $i, j = 1, \ldots, n$, are components of the matrix derivative $\frac{\partial \mathcal{L}}{\partial \mathbf{K}}$, for which the calculations of matrix differential calculus (Magnus and Neudecker, 2007) together with (15) and (41) yield

$$\frac{\partial \mathcal{L}}{\partial \mathbf{K}} = \frac{1}{2}\left(\mathbf{K}^{-1}\mathbf{y}\mathbf{y}^\top\mathbf{K}^{-1} - \mathbf{K}^{-1}\right). \tag{46}$$

Random Forests

Random forest by Breiman (2001) is a regression model formed as an ensemble of decision trees (Breiman, 1984). sampling strategy in the input space to the resulting landscape features and, consequently, to their relationship with the perfomance of the considered models and their various settings.

DECISION TREES Since Breiman's CART (Breiman, 1984), decision trees have been developed to a wide spectrum of variants (Criminisi et al., 2011). In this thesis, we have considered only binary regression trees where each internal node has two outcoming edges. In such regression trees, each observation $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in \mathbb{R}^D$ passes through a series of binary split functions h associated with internal nodes and arrives in the leaf node containing a predictor model p. Binary split function associated with the node i determining whether $\mathbf{x}$ belongs to its left or right child can be formulated as $h(\mathbf{x}, \theta_i) \in \{0, 1\}$, where θ_i are parameters 0 denotes passing to the left child (L), and 1 passing to the right child (R). A predictor model $p(\mathbf{x}) \in \mathbb{R}$ is represented by a simple regression model (e. g., constant, linear, quadratic) utilized for the prediction of the function value y.

A split $h(\mathbf{x}, \theta_i)$ together with its parameters θ_i are recursively chosen to maximize the split gain I achieved by splitting the data $\mathcal{S}$ to left and right nodes $\mathcal{S}_L$ and $\mathcal{S}_R$. The gain is calculated as

$$I = \operatorname{err}(\mathcal{S}) - \sum_{j \in \{L,R\}} \frac{|\mathcal{S}_j|}{|\mathcal{S}|} \operatorname{err}(\mathcal{S}_j), \tag{47}$$

where split gain function err represent some error measure (see the paragraph below).

There are several stopping criteria for growing individual branches or the tree as a whole: the tree reaches an allowed maximum number of levels or maximum number of splits, the value of gain function decreases below the user-defined threshold, a node contains at least the defined number of points and any additional split would violate it. To avoid over-fitting of a decision tree, one can use *tree pruning* (Breiman, 1984), which reduces the complexity of the final tree and improves predictive accuracy in general. To select the best level of pruning, predictive accuracy can be estimated using cross-validation.

SPLIT FUNCTIONS Traditional CART by Breiman (1984) is based on searching *axis-parallel* hyperplanes. To find the splitting hyperplane, the value of each training point $\mathbf{x} = (x_1, \ldots, x_D)$ in dimension $\mathbf{x}_d, d \in \{1, \ldots, D\}$ is considered as threshold for the dimension d defining the candidate hyperplane. The full-search through all dimensions is done and the splitting hyperplane with the highest gain is selected.

In SECRET introduced by Dobra and Gehrke (2002), the expectation-maximization algorithm for *Gaussian mixtures* is utilized to find two clusters and the regression task is transformed into classification based on assignments of points to these clusters. Splitting oblique hyperplane is provided through linear or quadratic discriminant analysis.

Deterministic *hill-climbing* with effective randomization in the algorithm OC1 presented by Murthy et al. (1994) is employed to find a most suitable linear hyperplane. Split-finding starts with a random hyperplane or with a good axis-parallel hyperplane found similarly to CART and deterministically perturbates the hyperplane's direction in each axis to maximize the split gain. Once no improvement is possible, a number of random jumps is performed as an attempt to escape from local optima. If some random jump succeeds, deterministic perturbation is performed again.

Hinton and Revow (1996) used the *pairs of points* to define a projection for splitting the input space. For each pair of points a normal vector defining a direction is constructed. The rest of training points is projected onto this vector and the projected values are utilized as thresholds defining hyperplanes orthogonal to constructed normal vector. deciding to which side the point belongs. To reduce complexity, only the threshold halfway between the defining pair can be considered.

A nonparametric function estimation method called SUPPORT by Chaudhuri et al. (1994) is based on the analysis of *residuals* after regression to find a split. At the start polynomial regression is performed on the training data. The points under the curve (negative residuals) present the first class, and the rest of points (positive or zero residuals) presents the second class. Afterwards, distribution analysis is applied to find a split.

SPLIT GAIN FUNCTIONS The original computation of gain used by Breiman (1984) on dataset $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ was through *mean-squared error* (MSE)

$$\text{err}_{\text{MSE}}(\mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \,, \tag{48}$$

where $\hat{y}_i$ is predicted value of point $\mathbf{x}_i$.

Considering the fact that each leaf contains not only a constant, model predicted values can also form the following *variance* gain function:

$$\text{err}_{\sigma^2}(\mathcal{S}) = \frac{1}{n^2} \sum_i \sum_{j>i} (\hat{y}_i - \hat{y}_j)^2 \,. \tag{49}$$

Criminisi et al. (2011) utilizes for gain calculations the *estimate of differential entropy* of an unknown continuous distribution

$$\text{err}_{\text{EDE}}(\mathcal{S}) = \sum_{i=1}^{n} \ln |\mathbf{C}_{\mathbf{y}}(\mathbf{x}_i)| \tag{50}$$

denoting as $\mathbf{C}_{\mathbf{y}}(\mathbf{x}_i)$ the covariance matrix of an estimated Gaussian distribution of linear model parameters fitted to $\mathcal{S}$. This function can be further simplified using approximated diagonal covariance matrix, where the only non-zero elements are variances s^2 of predicted values $\hat{y}$

$$\mathrm{err}_{\mathrm{EDE}}(\mathcal{S}) = 1 + \ln(2\pi) + \frac{1}{2n}\sum_{i=1}^{n} \ln s_i^2 \,. \tag{51}$$

Ahmed and Gokhale (1989) derived the *uniformly minimum-variance unbiased estimator* of entropy:

$$\mathrm{err}_{\mathrm{UMVUE}}(\mathcal{S}) = \frac{1}{2}\ln(e\pi) + \frac{1}{2}\ln(|\mathbf{y}\mathbf{y}^\top|) + \frac{1}{2}\psi\left(\frac{n}{2}\right) , \tag{52}$$

where ψ is the digamma function defined as $\psi(a) = \frac{\mathrm{d}\ln\Gamma(a)}{\mathrm{d}\,a}$.

For a non-parametric estimate of entropy, Nowozin (2012) suggests a *1-nearest neighbor estimator* by Beirlant et al. (1997)

$$\mathrm{err}_{\mathrm{NN}}(\mathcal{S}) = \frac{1}{n}\sum_{i=1}^{n} \ln \rho_i + \ln(n-1) + \gamma + 2 \,, \tag{53}$$

where $\rho_i = \min_{j\in\{1,\dots,n\}\setminus\{i\}} |y_j - y_i|$ is the distance to the nearest neighbor and $\gamma \approx 0.5772$ is the Euler-Mascheroni constant.

It is important to note that the $\mathrm{err}_{\mathrm{EDE}}$ and $\mathrm{err}_{\mathrm{UMVUE}}$ assume normal distribution of predicted values. Therefore, the leaf regressor should ensure normality of its output. Thus, utilization of those two functions in non-linear regression is limited, e. g., quadratic regression would require functions assuming χ^2-distribution.

ENSEMBLE METHODS As to the training of random forests, the most common approaches to it are *bagging* (Breiman, 1996) and *boosting* (Breiman, 1998). The most popular methods for creating ensembles of trees can be divided into two groups according to the order of learning individual trees:

- ◇ *sequential* where the trees are trained sequentially considering performance of the previously built trees (e. g., *boosting* by Breiman (1998)),
- ◇ and *paralel* where the trees are generated independently in paralel (e. g., *bagging* by Breiman (1996)).

In the following paragraphs, we will pay attention only to bagging and boosting method as the main representatives of ensemble learning methods.

Bagging Parallel learning of decision trees in bagging enables exploiting the independence between the individual trees. An important aspect of bagging are random differences between individual trees of the ensemble. This increases robustness and improves generalization of the prediction. There are three popular techniques of bagging: random sampling of training data set with replacement, using only a small subset of all possible parameter values while training, and making available only subsets of input features. Also, there's a possibility to combine the two of them.

For a given dataset of N points $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the overall forest prediction of the bagged forest $\mathcal{F}$ is provided by averaging all tree predictions.

$$\hat{y}_i = \sum_{f\in\mathcal{F}} f(\mathbf{x}_i) \,. \tag{54}$$

This form of prediction implies the larger the ensemble the greater robustness to noise.

Boosting Sequential learning of decision trees in boosting enables exploiting the dependence between the individual trees. In recent years, the *gradient tree boosting* (Friedman, 2001) has become very successful. Therefore, we will focus on this boosting method.

To learn a boosted forest, the model has to be trained additively. Let $\hat{y}_i^{(t)}$ be the prediction of the i-th point of the t-th tree. The t-th tree f_t is obtained in the t-th iteration of the boosting algorithm through optimization of the following regularized objective function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) + \Omega(f_t), \quad \Omega(f) = \gamma T_f + \frac{1}{2}\lambda \left\| w_f \right\|^2, \tag{55}$$

where l is a differentiable convex loss function $l : \mathbb{R}^2 \to \mathbb{R}$, T_f is the number of leaves in a tree f, w_f are weights of its individual leaves. The regularization term Ω is used to control the complexity of model through penalization constants γ and λ. From (55) can be derived (see Chen and Guestrin (2016) for details) that the gain of split considering splitting data $\mathcal{S}$ into sets $\mathcal{S}_L$ and $\mathcal{S}_R$ according to the nodes where they belong will be

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left(\frac{\left(\sum_{y\in\mathcal{S}_L} g(y)\right)^2}{\sum_{y\in\mathcal{S}_L} h(y) + \lambda} + \frac{\left(\sum_{y\in\mathcal{S}_R} g(y)\right)^2}{\sum_{y\in\mathcal{S}_R} h(y) + \lambda} - \frac{\left(\sum_{y\in\mathcal{S}} g(y)\right)^2}{\sum_{y\in\mathcal{S}} h(y) + \lambda} \right) - \gamma, \tag{56}$$

where $g(y) = \partial_{\hat{y}^{(t-1)}} l(y, \hat{y}^{(t-1)})$ and $h(y) = \partial^2_{\hat{y}^{(t-1)}} l(y, \hat{y}^{(t-1)})$ are first and second order derivatives of the loss function. Using equation (47), one can define *gradient error measure* of node j:

$$\text{err}_{\text{Grad}}(\mathcal{S}_j) = -\frac{1}{2}\Big(\sum_{y\in\mathcal{S}_j} g(y)\Big)^2 \Big(\sum_{y\in\mathcal{S}_j} h(y) + \lambda\Big)^{-1}. \tag{57}$$

The overall boosted forest prediction is again obtained through averaging according to (54). In this case, however, each leaf and tree has a different weight. As another prevention of overfitting, a technique scaling weights of recently added trees by a constant μ called *shrinkage* (Friedman, 2002) can be used. Moreover, utilizing sampling strategies from bagging is also popular.

2.3.2 *Evolution Control*

Evolution control (EC) determines the subset of individuals to be evaluated using the original fitness function; the remaining individuals are evaluated by the surrogate model. Following a terminology by Jin (2005), there are two main classes of evolution control : *individual-based*, where a fraction of each population is controlled, and *generation-based*, where all individuals in one generation are either evaluated using the model or using the original fitness function.

Let us denote λ to be the size of population and $\varepsilon \in [0, 1]$ the fraction of controlled individuals in *individual-based* control. Thus there will be $\varepsilon\lambda$ individuals in controlled population. First, an extended population $\alpha(1-\varepsilon)\lambda$ is created where $\alpha > 1$ is an extension parameter. In the next step, $(1-\varepsilon)\lambda$ individuals are chosen according to specific criterion. Those individuals are subsequently evaluated with the original fitness function. The size of λ, ε, and the criterion how to select individuals to be evaluated with the original fitness function are open and problem dependent questions (Büche et al., 2005).

There are several posibilities how to select the individuals for the original fitness reevaluation. The most straightforward selection of individuals is based on the model fitness function values. The selection criterion can choose individuals with the best fitness of the entire extended population or cluster all individuals first and choose representatives of each cluster afterwards. Several different strategies how to choose representatives can be applied. Another popular criterion is using unexplored regions, which are areas where no points have been sampled yet. Here, Gaussian processes and random forest models also provide an uncertainty measure how the prediction of the fitness function value is precise.

Generation-based control distinguishes between two types of generations: generations where all individuals are evaluated with the original fitness function and generations in which they are evaluated with the model fitness. General generation control approach trains the model in the original fitness generation after the evaluation. And this model is subsequently used for prediction in the following generations using only model fitness evaluation. Adaptive evolution control can be used; for example, Loshchilov et al. (2012) counts deviation between the original and the model fitness function and then decides whether to evaluate with the original fitness or with the model. A simpler approach is to keep the number of model fitness generations constant, which can, however, easily slow-down the convergence of the algorithm.

2.3.3 *Surrogate Versions of CMA-ES*

In this section, we give some details about several surrogate-model versions of the CMA-ES that we consider the most important.

MA-ES

To the best of our knowledge, the first combination of the CMA-ES optimizer and a Gaussian process model was in the *Metamodel-Assisted Evolution Strategy* (MA-ES) proposed by Ulmer et al. (2003). According to Jin's terminology, the algorithm employs individual-based evolution control (see Section 2.3.2). The algorithm pre-selects points based on one of two criteria: the best GP mean prediction, or the *Probability of Improvement* (PoI)—the probability that the considered point's function value will drop below some value, usually the so-far achieved optimum $\mathfrak{b}_{\min}$ (see Criteria paragraph in Section 4.4). The authors have shown that the latter criterion, which uses the GP's prediction of uncertainty, performs better especially on multimodal functions like Ackley's or Griewank. The same algorithm name was used for the GP-surrogate evolutionary strategy (not necessarily the CMA-ES) by Emmerich et al. (2002, 2006) who also showed superior performance of the criteria exploiting the GP uncertainty on multimodal functions.

LS-CMA-ES

The *Least-Square CMA-ES* (LS-CMA-ES), introduced by Auger et al. (2004), uses a quadratic model approximation of the fitness trained on the recent $O(n^2)$ points. This approximation is shown to provide a faster update of the CMA-ES covariance matrix when applied to fitness functions similar to ellipsoid. The algorithm also automatically controls whether to use the model for the update or not.

GPOP

Büche et al. (2005) combined GP surrogate model for the CMA-ES in their *Gaussian Process Optimization Procedure* (GPOP). It is a Bayesian optimization algorithm which additionally restricts the search around the current best point $\hat{\mathbf{x}}_{\text{opt}}$. In each iteration, the GPOP constructs a GP model of the local neighborhood of $\hat{\mathbf{x}}_{\text{opt}}$, and the CMA-ES is then used to locate the optimum of this model for evaluation with the original fitness. The CMA-ES optimizes functions defined as $(\hat{y}(\mathbf{x}) - \alpha\hat{s}(\mathbf{x}))$ where $\hat{y}(\mathbf{x})$ and $\hat{s}(\mathbf{x})$ are the mean and standard deviation of the GP prediction in $\mathbf{x}$ respectively. The authors suggest using four different $\alpha = 0, 1, 2, 4$ in each generation: while $\alpha = 0$ exploits the current best-predicted point, $\alpha = 4$ explores mainly uncertain regions of the search space.

Local Meta-model CMA-ES

An effective combination of building local surrogate models and controlling changes in population ranking after fitness function evaluation is incorporated in the *local meta-model CMA-ES* (lmm-CMA), proposed by Kern et al. (2006) and later improved in (Auger et al., 2013; Bouzarkouna et al., 2010, 2011).

Locally weighted regression (LWR) by Atkeson et al. (1997) is employed to build an individual surrogate model $f_{\mathcal{M}}$ for every offspring to be predicted, called query point $\mathbf{q} \in \mathbb{R}^D$, using a set of k_{nn} training points nearest to $\mathbf{q}$ according to some distance d, yielding $d(\mathbf{q}, \mathbf{x}_i)$, $i = 1, \ldots, k_{\text{nn}}$. The number k_{nn} was experimentally chosen by Kern et al. (2006) as $k_{\text{nn}} = D(D+3)+2$. The distance d is the Mahalanobis distance associated to the covariance matrix $\sigma^2\mathbf{C}$ (using the CMA-ES step-size σ and matrix $\mathbf{C}$) for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^\top (\sigma^2\mathbf{C})^{-1}(\mathbf{x}-\mathbf{y})}\,. \tag{58}$$

The considered surrogate model $f_{\mathcal{M}}$ is full quadratic:

$$f_{\mathcal{M}}(\tilde{\mathbf{x}}, \boldsymbol{\beta}) = \tilde{\mathbf{x}}^\top \boldsymbol{\beta}\,, \tag{59}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{\frac{D(D+1)}{2}-1}$ and $\tilde{\mathbf{x}} = \left(x_1^2, \ldots, x_D^2, x_1x_2, \ldots, x_{D-1}x_D, x_1, \ldots, x_D, 1\right)$. The following weighted least square error is minimized w.r.t. the parameters $\boldsymbol{\beta}$ of the local model $f_{\mathcal{M}}$

$$\text{err}_{\text{WLSE}}(\mathbf{q}) = \sum_{i=1}^{k_{\text{nn}}} K\left(\frac{d(\mathbf{x}_i, \mathbf{q})}{h}\right)\left(f_{\mathcal{M}}(\mathbf{x_i}, \boldsymbol{\beta}) - y_i\right)^2\,, \tag{60}$$

where $K(.)$ is a kernel function and h is a parameter called *local bandwidth*. Such a distance weighting corresponds to requiring the local model to approximate nearby points more precisely than distant points. Therefore, we can minimize (60) by solving the equation

$$\left((\mathbf{W}\tilde{\mathbf{X}})^\top \mathbf{W}\tilde{\mathbf{X}}\right)\boldsymbol{\beta} = (\mathbf{W}\tilde{\mathbf{X}})^\top \mathbf{W}\mathbf{y}\,, \tag{61}$$

where $\mathbf{W} = \text{diag}\,\sqrt{K(d(\mathbf{x}_i, \mathbf{q})/h)}$, $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_{k_{\text{nn}}})^\top$, and $\mathbf{y} = (y_1, \ldots, y_{k_{\text{nn}}})^\top$. The kernel function used by Auger et al. (2013) was bi-quadratic $K(\xi) = (1-\xi^2)^2$; however, in the original lmm-CMA (Kern et al., 2006), the kernel was chosen to be partially bi-quadratic:

$$K(\xi) = \begin{cases} (1-\xi^2)^2 & \text{if } \xi < 1 \\ 0 & \text{otherwise}\,. \end{cases} \tag{62}$$

Algorithm 2 Ranking prediction in the lmm-CMA variant proposed by Auger et al. (2013)

Input: population-size λ, new population $\{\mathbf{x}_l\}_{l=1}^{\lambda}$, original fitness function bb, archive $\mathcal{A}$, minimal number of points for model training n_{init}, increment of number of points for model training n_b, number of nearest neighbors k_{nn}, quality threshold t_q

$\{f_{\mathcal{M}}^{(l)}\}_{l=1}^{\lambda} \leftarrow \text{LWR}(\{\mathbf{x}_l\}_{l=1}^{\lambda}, k_{\text{nn}}, \mathcal{A})$ {*model building*}
$\text{ranking}_i^{\mu} \leftarrow$ rank μ best $\mathbf{x}_l$ according to $f_{\mathcal{M}}^{(l)}(\mathbf{x}_l)$ {*ranking*}
$y_l \leftarrow \text{bb}(\mathbf{x}_l) \qquad l \in \{n_{\text{init}} \text{ best ranks}\}$ {*fitness evaluating*}
$\mathcal{A} = \mathcal{A} \cup \{(\mathbf{x}_l, y_l)\}_{l \in \{n_{\text{init}} \text{ best ranks}\}}$
$\{f_{\mathcal{M}}^{(l)}\}_{l=1}^{\lambda} \leftarrow \text{LWR}(\{\mathbf{x}_l\}_{l=1}^{\lambda}, k_{\text{nn}}, \mathcal{A})$ {*model building*}
$\text{ranking}_0^{\mu} \leftarrow$ rank μ best $\mathbf{x}_l$ according to $f_{\mathcal{M}}^{(l)}(\mathbf{x}_l)$ {*ranking*}
$\text{err}_{\text{model}} \leftarrow +\infty$
$i = 0$
while $\exists l \in \{1, \ldots, \lambda\}\ (\mathbf{x}_l, y_l) \notin \mathcal{A}$ **and** $\text{err}_{\text{model}} > t_q$ **do**
  $i \leftarrow i + 1$
  $y_l \leftarrow \text{bb}(\mathbf{x}_l) \qquad l \in \{n_b \text{ best unevaluated}\}$ {*fitness evaluating*}
  $\mathcal{A} = \mathcal{A} \cup \{(\mathbf{x}_l, y_l)\}_{l \in \{n_b \text{ best unevaluated}\}}$
  $\{f_{\mathcal{M}}^{(l)}\}_{l=1}^{\lambda} \leftarrow \text{LWR}(\{\mathbf{x}_l\}_{l=1}^{\lambda}, k_{\text{nn}}, \mathcal{A})$ {*model building*}
  $\text{ranking}_i^{\mu} \leftarrow$ rank μ best $\mathbf{x}_l$ according to $f_{\mathcal{M}}^{(l)}(\mathbf{x}_l)$ {*ranking*}
  $\text{err}_{\text{model}} \leftarrow \sum_{1 \le j \le \mu} |\text{ranking}_{i-1}^{\mu}(j) - \text{ranking}_i^{\mu}(j)|$
end while
if $i > 2$ **then**
  $n_{\text{init}} = \min(\lambda, n_{\text{init}} + n_b)$
end if
if $i < 2$ **then**
  $n_{\text{init}} = \max(0, n_{\text{init}} - n_b)$
end if

Output: ranking_i^{μ}

In lmm-CMA, the model quality is controlled by an *approximate ranking procedure* by Runarsson (2004), which suggests evaluating a portion of the best individuals on the original fitness function bb if the ranking of the fraction of the offspring remains unchanged in two consecutive iterations. The algorithm is using an archive $\mathcal{A}$ of points evaluated on the original fitness function bb. The original fitness evaluation phase of the CMA-ES is replaced by the ranking prediction outlined in Algorithm 2. The shown algorithm was proposed by Auger et al. (2013), being numerically more stable than the original by Kern et al. (2006). The standard CMA-ES iterations run until the minimal number of points required to start ranking prediction equals to the number of free parameters of one meta-model $m_{\text{dim}} + 1$, where $m_{\text{dim}} = D(D+3)/2 + 1$. The initial values of ranking prediction parameters suggested by Auger et al. (2013) are $n_{\text{init}} = 1$, $n_b = \max\{\lceil \frac{\lambda}{20}, 1 \rceil\}$, quality threshold $t_q = \frac{\lambda^2}{20}$, and $k_{\text{nn}} = \lceil \min\{2\, m_{\text{dim}}, \sqrt{|\mathcal{A}|\, m_{\text{dim}}}\} \rceil$.

In a version nlmm-CMA by Bouzarkouna et al. (2010), the condition used to stop evaluating new points using original fitness was improved to keep the speed-up for larger populations than the CMA-ES default.

The lmm-CMA is analysed in more detail in Section 4.8.1.

Kriging metamodeling CMA-ES

Kruisselbrink et al. (2010) used GPs to handle robustness in optimization of noisy functions. Their *Kriging metamodeling based CMA-ES* evaluates the population with the noisy original function and then constructs a separate GP/kriging model for each such point in order to estimate its true value without noise. The algorithm uses the CMA-ES covariance matrix $\mathbf{C}$ for the transformation of the input space. Apart from a procedure for selecting training points for the GP from an archive, the authors also suggest which points to take for an additional expensive evaluation if there are not enough points for the GP model building. The kriging method is shown to approximate the noisy fitness much better than two compared methods based on a simple (re-)evaluation of the points during CMA-ES optimization.

Ensemble of Local Gaussian Process Models

The article by Lu et al. (2013) presents another surrogate-assisted CMA-ES algorithm that employs an ensemble of local Gaussian process models. The algorithm uses the GP models to pre-screen a larger population ($\lambda = 25$, $\mu = 5$) and to select the best 6 points for expensive re-evaluation according to the same criterion as Büche et al. (2005) with $\alpha = 2$. Although the results in the article seem rather promising, we were not able to reproduce them using the provided source code.

^{s*}ACM-ES-k

Loshchilov et al. (2012) combined the ability of Ranking SVR (Herbrich et al., 1999) to preserve the CMA-ES invariance properties with an adaptation of surrogate-model hyperparameter values during the search in their ^{s*}ACM-ES. An extension of that algorithm using a more intensive exploitation is called ^{s*}ACM-ES-k (Loshchilov et al., 2013b).

Ranking SVR (Herbrich et al., 1999) is a variant of SVR for ordinal regression based on large margins between individual rank boundaries.

The ^{s*}ACM-ES-k starts with evaluating g_{start} generations using the original fitness. Then it iterates through the following steps:

1. *train* a surrogate model with parameters θ using the points evaluated with the original fitness;
2. *optimize* the surrogate model by the CMA-ES for g_m generations with population size $\lambda = k_\lambda \lambda_{\text{default}}$ and the number of parents $\mu = k_\mu \mu_{\text{default}}$, where $k_\lambda, k_\mu \geq 1$;
3. *evaluate* the original function f on the CMA-ES generated offspring using $\lambda = \lambda_{\text{default}}$ and $\mu = \mu_{\text{default}}$;
4. *calculate* the model error and subsequently the new g_m using ranks of the original and model evaluations of the last generation;
5. *search* the parameter space of the surrogate model by the CMA-ES to find the most convenient settings θ_{new} for the next-generation model, using the model error as a fitness function.

As a prevention against algorithm divergence, $k_\lambda > 1$ is used only in the case of $g_m \geq g_{m_\lambda}$, where g_{m_λ} denotes the number of generations suitable for effective exploitation using the model.

In (Loshchilov et al., 2013a), the ^{s*}ACM-ES-k version using BIPOP-CMA-ES, called BIPOP-^{s*}ACM-ES-k, and a hybrid of BIPOP-^{s*}ACM-ES-k, the *select the easiest point* (STEP) method by Swarzberg et al. (1994) and the NEWUOA algorithm by Powell (2006), called HCMA, were proposed.

EGO-CMA

Mohammadi et al. (2015) combined GPs and the CMA-ES in their EGO-CMA. It starts as the EGO algorithm (see Subsection 2.4) and switches into the CMA-ES after a small number of EGO iterations. Hence, it is not a surrogate model in the proper sense of repeatedly switching between the evaluation with the original fitness and with the model. The core contribution of the article lies in the sophisticated initialization of the CMA-ES covariance matrix and step-size based on the learned parameters of the EGO's Gaussian process. The authors show interesting preliminary results on the 5-dimensional Sphere and Ackley functions which we were also not able to reproduce due to the incompleteness of the provided source code.

SAPEO

Another combination of GP model and the CMA-ES algorithm is the *Surrogate-Assisted Partial Order-Based Evolutionary Optimization Algorithm* (SAPEO) by Volz et al. (2017). The algorithm uses several dominance relations that combine the GP predicted mean and variance to induce a partial order on the points in a CMA-ES population, usually using confidence intervals. If this partial order is able to identify the ranking of the μ best points with a given probability, the algorithm proceeds without any further fitness evaluations. Otherwise, the SAPEO tries to choose its another dominance relation or original-evaluates some of the points if necessary.

lq-CMA-ES

The most recent single-objective surrogate-assisted CMA-ES algorithm we are aware of is the *linear-quadratic Global Surrogate Assisted CMA-ES* (lq-CMA-ES) by Hansen (2019) switching between the linear or diagonal-quadratic or full quadratic model according to the number of available training data.

Similarly to the previous surrogate-assisted versions of the CMA-ES, lq-CMA-ES stores already evaluated points. Nevertheless, in this case the observations are stored in a queue $\mathcal{Q}$. In each iteration, a surrogate model $f_{\mathcal{M}}$ is built utilizing points from $\mathcal{Q}$. Then the entire population is evaluated using the built model. After that, a small number of model-best solutions is selected from the population, evaluated on f, then sorted and enqueued (the best solution is enqueued last). When the maximum queue size is exceeded, the oldest elements are dropped. The number of individuals selected for evaluation using the fitness is repeatedly increased and the model repeatedly trained incorporating newly evaluated points until the Kendall's rank correlation coefficient τ between the model-predicted and fitness-evaluated ordering of current population exceeds a specified constant (set to 0.85 by Hansen (2019)). The procedure is outlined in Algorithm 3.

The resulting ordering of the current population is completely determined using the surrogate model only or the fitness function only considering that all points would be evaluated by the original fitness. Thus, fitness and model values cannot be mixed in the returned ordering. After returning the values to the CMA-ES, the model optimum is injected to be used as candidate direction in the following generation according to Hansen (2011).

Algorithm 3 Evolution control in the lq-CMA-ES by Hansen (2019)

Input: $\mathcal{Q} = \{(\mathbf{z}_1, \text{bb}(\mathbf{z}_1)), \ldots, (\mathbf{z}_l, \text{bb}(\mathbf{z}_l))\}, l \in \{1, \ldots, \max(\lambda, D^2 + 3D + 2)\}$ (model-training queue), $f_{\mathcal{M}}$ (model), $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\lambda\}$ (population), bb (fitness function)

$k \leftarrow \lfloor 1 + \max(0.02\lambda, 4 - |Q|) \rfloor$ {*incrementing evaluations*}
while $|\mathcal{P}| > 0$ **do**
  train $f_{\mathcal{M}}$ on k points from $\mathcal{Q}$
  drop the $k - (\lambda - |\mathcal{P}|)$ $f_{\mathcal{M}}$-best elements from $\mathcal{P}$ into $\mathcal{Q}$
  sort the newest $\min(k, \lambda)$ elements in Q w.r.t. their bb values
  $k_\tau \leftarrow \max(15, \min(1.2k, 0.75\lambda))$
  $(\mathbf{z}_j)_{j=1}^{k_\tau} \leftarrow$ the last k_τ points in $\mathcal{Q}$
  if Kendall-τ $([f_{\mathcal{M}}(\mathbf{z}_i)]_i, [\text{bb}(\mathbf{z}_i)]_i) \geq 0.85, i \in \{1, \ldots, k_\tau\}$ **then**
    break while
  end if
  $k \leftarrow \lceil 1.5k \rceil$
end while
if $|\mathcal{P}| > 0$ **then**
  return $\mathcal{Q}$, $f_{\mathcal{M}}(\mathbf{x}_1)$, $\ldots$, $f_{\mathcal{M}}(\mathbf{x}_\lambda)$ all offset by $\min_{\mathbf{x} \in \text{last } k \text{ elements of } \mathcal{Q}}(\text{bb}(\mathbf{x})) - \min_{i=1,\ldots,\lambda}(f_{\mathcal{M}}(\mathbf{x}_i))$
else
  return $\mathcal{Q}$, $\text{bb}(\mathbf{x}_1), \ldots, \text{bb}(\mathbf{x}_\lambda)$
end if

The employed surrogate model switches between tree kinds of polynomial model: linear, pure quadratic, and full quadratic. The lq-CMA-ES switches to the next polynomial model when the number of training points exceeds the degrees of freedom of the next model plus 10%.

The lq-CMA-ES is analysed in more detail in Section 4.8.1.

2.4 Other Approaches to Black-Box Optimization

Evolutionary algorithms and their surrogate-assisted versions are not the only algorithms capable to optimize black-box functions. In this section, we will briefly introduce 3 algorithms which do not follow the CMA-ES approach.

Nelder-Mead

Nelder and Mead (1965) proposed a *downhill simplex method* operating only on the ranking of the problem solutions. Therefore, the algorithm is invariant under the same order-preserving transformations of the fitness function as the CMA-ES. As opposed to CMA-ES (and all stochastic algorithms), the original Nelder-Mead method is strictly deterministic, i. e., it does not sample.

The Nelder-Mead algorithm is based on the usage of $(D+1)$-simplexes of solution points, where D is the dimension of the search space. In each iteration, a new simplex point is computed as a linear combination of the worst solution comprised in the simplex and the center of the remaining points. The new point then updates the simplex according to its

function value using the following operations: *reflection*, *expansion*, *contraction*, and *multiple contraction* (see Algorithm 4). The uncorrected sample standard deviation of current simplex fitness values $\sigma\left(\{\mathfrak{b}(\mathbf{x}_i)\}_{i=0}^{D}\right)$ is used as the stopping criterion (step 2).

This successful optimizer has been used for decades (it is, for example, the standard part of Matlab), but due to its deterministic nature, it is unable to escape from local optimum. Therefore, a few restart strategies for global optimization problems have been proposed and benchmarked (Hansen, 2009b; Luersen and Le Riche, 2004).

Brent's method

A popular algorithm for univariate black-box optimization problems was introduced by Brent (1973). The algorithm is a combination of root bracketing, bisection, and inverse quadratic interpolation. Generally, three points bracketing a local optimum are interpolated by a quadratic function. The minimum of the quadratic function satisfying some specific conditions is proceeded to the next iteration for interpolation. A bisection step to compute a new point is initiated if the conditions are violated.

The algorithm is convenient for the local search in one dimension. For multidimensional purpose it is combined with optimization methods such as *line search*. The strength of univariate algorithms is in separable problems where the problem can be decomposed into D univariate problems and optimize one after another in a sequence. To ensure convergence not only on unimodal functions but also on multimodal functions, the Brent local search method and a global search STEP algorithm by Swarzberg et al. (1994) were combined by Baudiš and Pošík (2015) in hybrid Brent-STEP method.

Bayesian optimizers

Bayesian optimization algorithms introduced by Mockus et al. (1978) in *sequential model based optimization* of expensive functions constitute a family of algorithms that use a Bayesian probabilistic model of the fitness to iteratively sample new promising points from the input space. The model has to express a probability distribution of the response variable, and this distribution is then exploited for deciding the new points. The decision which point to take for the next iteration is based on an *acquisition function* that quantify the interestingness of points from the input space. The expected improvement, probability of improvement, or a quantile of the predictive distribution is used most often.

Great success of Bayesian optimization algorithms was brought by Jones et al. (1998) with their Gaussian-process-based *Efficient Global Optimization* (EGO) algorithm. The EGO and its derivatives find their applications especially in the very expensive scenarios where the allowable budget of function evaluations is not more than few hundreds.

One of succesors of EGO the *Sequential Model-based Algorithm Configuration* (SMAC) method introduced by Hutter et al. (2011) was originally designed to optimize algorithm parameters. It fits surrogate models of algorithm settings in a parameter space and utilizes those models to make decisions about which settings to investigate.

To minimize function $\mathfrak{b} : \boldsymbol{\Theta} \mapsto \mathbb{R}$, where $\boldsymbol{\Theta}$ is the space of algorithm parameters, SMAC iterates over the following steps:

1. *build* a model $f_{\mathcal{M}}$ based on random forest predicting a probability distribution of $\mathfrak{b}$ using previously evaluated data points $(\boldsymbol{\theta}, \mathfrak{b}(\boldsymbol{\theta}))$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$;

Algorithm 4 Nelder-Mead downhill simplex method

Input: $\{\mathbf{x}_i\}_{i=0}^{D}$ (initial simplex), $\mathbb{bb}$ (original fitness function),
$\alpha > 0$ (reflection coefficient), $\beta \in (0, 1)$ (contraction coefficient),
$\gamma > 1$ (expansion coefficient), ε (stopping-criterion constant)

$k \leftarrow 0$
while $\sigma\left(\{\mathbb{bb}(\mathbf{x}_i)\}_{i=0}^{D}\right) > \varepsilon$ **and** $k < k_{\max}$ **do**
  $h \leftarrow \operatorname{argmax}_{i \in \{0,\ldots,D\}} \mathbb{bb}(\mathbf{x}_i)$
  $l \leftarrow \operatorname{argmin}_{i \in \{0,\ldots,D\}} \mathbb{bb}(\mathbf{x}_i)$
  $\bar{\mathbf{x}} \leftarrow \sum_{i=0}^{D} \mathbf{x}_i$
  $\mathbf{x}' \leftarrow (1 + \alpha)\bar{\mathbf{x}} - \alpha \mathbf{x}_h$
  if $\mathbb{bb}(\mathbf{x}') < \mathbb{bb}(\mathbf{x}_l)$ **then**
    $\mathbf{x}'' \leftarrow (1 + \gamma)\bar{\mathbf{x}} - \gamma \mathbf{x}_h$
    if $\mathbb{bb}(\mathbf{x}'') < \mathbb{bb}(\mathbf{x}_l)$ **then**
      $\mathbf{x}_h \leftarrow \mathbf{x}''$ *{expansion}*
    else
      $\mathbf{x}_h \leftarrow \mathbf{x}'$ *{reflection}*
    end if
  else if $\mathbb{bb}(\mathbf{x}') > \mathbb{bb}(\mathbf{x}_i), \forall i \neq h$ **then**
    if $\mathbb{bb}(\mathbf{x}') \leq \mathbb{bb}(\mathbf{x}_h)$ **then**
      $\mathbf{x}_h \leftarrow \mathbf{x}'$ *{reflection}*
    end if
    $\mathbf{x}'' \leftarrow \beta \mathbf{x}_h + (1 - \beta)\bar{\mathbf{x}}$
    if $\mathbb{bb}(\mathbf{x}'') > \mathbb{bb}(\mathbf{x}_h)$ **then**
      $\mathbf{x}_i \leftarrow \frac{\mathbf{x}_i + \mathbf{x}_l}{2}$ $\quad i \in \{0, \ldots, D\}$ *{multiple contraction}*
    else
      $\mathbf{x}_h \leftarrow \mathbf{x}''$ *{contraction}*
    end if
  else
    $\mathbf{x}_h \leftarrow \mathbf{x}'$ *{reflection}*
  end if
  $k \leftarrow k + 1$
end while

Output: $\mathbf{x}_l$

2. *compute expected positive improvement* $E(I(\boldsymbol{\theta})) = E(\max\{0, \text{b}_{\min} - \text{b}(\boldsymbol{\theta})\})$ for each parameter configuration $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ using the model $f_{\mathcal{M}}$, where $\text{b}_{\min}$ is the lowest fuction value evaluated so far;

3. *select* the most promising point $\boldsymbol{\theta}_{\max} = \operatorname{argmax}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} E(I(\boldsymbol{\theta}))$;

4. *evaluate* $\text{b}(\boldsymbol{\theta}_{\max})$ and store $(\boldsymbol{\theta}_{\max}, \text{b}(\boldsymbol{\theta}_{\max}))$.

To make SMAC more useful in continuous optimization, random forests were replaced by Gaussian processes as a surrogate model in SMAC-BBOB (Hutter et al., 2013).

2.5 EXPLORATORY ANALYSIS OF FITNESS LANDSCAPES

If we want to group investigated problems according to their similarities, it is very useful to characterize an unknown landscape of the objective function. Following the idea that methods often perform well on entire classes of similar problems rather than just on single problems, Mersmann et al. (2010) have shown that such knowledge can be very helpful.

Measures quantifying the most important kinds of information about the landscape were formulated by Mersmann et al. (2010): the degree of *multi-modality*, the underlying *global structure*, the *separability*, the *variable scaling*, the *search space homogeneity*, the *basin size homogeneity*, whether the function's landscapes posesses *plateaus*, and the *local to global optima contrast*. In addition, some *high-level* properties can be used, which however have the disadvantages of missing important information, and requiring knowledge about the whole problem (Kerschke, 2017b).

These issues were addressed by Mersmann et al. (2011) where the authors introduced *Exploratory Landscape Analysis* (ELA), which is an umbrella term for analytical, approximated and non-predictive methods originally developed for combinatorial optimization problems (Muñoz et al., 2015). An important step in the development of ELA was proposing six *low-level* easy to compute feature sets (Mersmann et al., 2011) each containing a number of individual features. These features are computable based on a sample of observations from the given problem, thus eliminating the impact of missclassification by the expert. Mersmann et al. (2011) have also shown that these low-level features relate well to the above mentioned *high-level* properties.

To formalize these low-level features, let us consider a sample set $\mathcal{S}$ of N pairs of observations in the context of continuous black-box optimization

$$\mathcal{S} = \left\{ (\mathbf{x}_i, y_i) \in \mathbb{R}^D \times \mathbb{R} \cup \{\circ\} \mid i = 1, \ldots, N \right\}, \tag{63}$$

where $\circ$ denotes missing y_i value (e. g., $\mathbf{x}_i$ was not evaluated yet). Then the sample set can be utilized to describe landscape properties using a *landscape feature*

$$\varphi : \bigcup_{N \in \mathbb{N}} \mathbb{R}^{N,D} \times (\mathbb{R} \cup \{\circ\})^{N,1} \mapsto \mathbb{R} \cup \{\pm\infty, \bullet\}, \tag{64}$$

where $\bullet$ denotes impossibility of feature computation.

The importance of low-level landscape features have been shown by Bischl et al. (2012) in the *algorithm selection problem* or *algorithm configuration problem* formulated by Rice (1976) which aims to find the best algorithm or algorithm configuration for a specific problem instance. Considering algorithm selection problem, fitness landscape analysis aims at characterizing the

landscape of a black-box function and deriving rules how those characteristics influence the performance of the optimization algorithm.

A large number of various fitness landscape analysis techniques have been proposed for continuous optimization in recent years. As mentioned above, Mersmann et al. (2011) proposed six feature sets representing different properties:

- ◇ *y-Distribution* set with measures related to the distribution of the objective function values,
- ◇ *Levelset* features capturing the relative position of each value with respect to quantiles of all objective values,
- ◇ *Meta-Model* features extracting the information from linear or quadratic regression models fitted to the sampled data,
- ◇ *Convexity* set describing the level of function landscape convexity,
- ◇ *Curvature* set with gradient and Hessian approximation statistics,
- ◇ and *Local Search* features related to local searches conducted from sampled points.

The last three feature sets require additional objective function evaluations.

The *cell-mapping* (CM) approach proposed by Kerschke et al. (2014) discretizes the input space to a user-defined number of blocks (i. e., cells) per dimension. Afterwards, the corresponding features are based on the relations between the cells and points within. Five cell-mapping feature sets were defined:

- ◇ *CM Angle* features extract information based on the location of the best and worst observation within a cell w.r.t. the corresponding cell center,
- ◇ *CM Convexity* estimate convexity of representative observations from three successive cells in each dimension,
- ◇ *CM Gradient Homogeneity* provide aggregated cell-wise information on the gradients between each point of a cell and its corresponding nearest neighbor,
- ◇ *Generalized CM* features are based on estimated transition probabilities of moving from one cell to one of its neighboring cells,
- ◇ *Barrier tree* features representing the local optima by tree leaves and landscape ridges by the branching nodes using probabilities from Generalized CM to build a so-called *barrier tree* by Flamm et al. (2002).

Additionally, two extra feature set taking into account division of the input space into blocks were proposed:

- ◇ *Linear Model* features by Kerschke (2017b) aggregating information about coefficients of linear models fitted in each cell and
- ◇ *SOO Tree* features by Derbel et al. (2019) based on trees mapping the search trajectory of an optimization heuristic.

It should be noted that cell-mapping approach is less useful in higher dimensions where the majority of cells is empty and feature computation can require a lot of time and memory.

Kerschke et al. (2015) proposed *nearest better clustering* (NBC) features based on the detection of funnel structures. The calculation of such features relies on the comparison of distances from observations to their nearest neighbors and their *nearest better neighbors*, which are the nearest neighbors among the set of all observations with a better objective value.

Lunacek and Whitley (2006) proposed the set of *dispersion features* comparing the dispersion among the data points and among subsets of these points from the sample set.

The *information content* features of a continuous landscape are derived in *Information Content of Fitness Sequences* approach by Muñoz et al. (2015) as the adaptation of methods for calculating of the information content of discrete landscapes.

Features measuring the proportion of principle components needed to explain a user-defined percentage of data's variance were presented in *principle component analysis* feature set by Kerschke (2017b).

Kerschke (2017b) also proposed the features providing basic information about the data such as number of points, boundaries or dimension ensembled in *Basic* feature set.

Kerschke et al. (2016) have shown $50D$ observations generated using *improved latin hypercube sampling* proposed by Beachkofski and Grandhi (2002) can be enough for certain high-level properties by means of numerical features.

A comprehensive survey of fitness landscape analysis methods can be found e. g., in (Muñoz et al., 2015; Pitzer and Affenzeller, 2012). An implementation of previously mentioned low-level features is available in the R-package `flacco` (Kerschke and Dagefoerde, 2017). A more detailed description of feature sets used in our research can be found in Appendix A.

As Kerschke (2017a) noted, most of the low-level features provide very useful information for analysis of problem landscapes and further research connected to it but do not provide intuitively understandable values. Moreover, some are even stochastic and thus should be computed multiple times and then appropriately aggregated.

Research into using fitness landscape features in connection with surrogate models is only starting. Yu et al. (2016) utilized *fitness distance correlation* for automatic selection between polynomial and RBF models and their settings as surrogates for a particle swarm optimization algorithm. Furthermore, such features have been investigated in connection with surrogate models in the context of black-box optimization using static settings only, where the model is selected once at the beginning of the optimization process (Saini et al., 2019). Moreover, the analysis of landscape features also had less attention so far than it deserves and not in the context of surrogate models (Renau et al., 2019, 2020).

2.6 Optimization Benchmarking

Comparison of different optimization algorithms can be based on a number of criteria. Practical applications are interested mainly in one of the two following or their combination: the quality of the found solution Δb, i. e., the minimal b-value distance from the global optimum b_{opt}, and the amount of time or other resources (energy, material, CPU time, etc.) needed to converge. As mentioned in Section 2.1, the resources in black-box optimization are frequently measured in the number of fitness function evaluations function evaluation (FE). In addition, for comparison accross different dimensions, the number of FEs is often divided by the dimension D of the input space FE$/D$.

2.6.1 *COCO framework*

In the following chapters, experimental algorithm comparisons are mostly based on the *COmparing Continuous Optimizers* (COCO) platform by Hansen et al. (2009a,b, 2012, 2021) which is quite successful in quantifying and comparing the performance of optimization algorithms in a scientifically decent and rigorous way. Is is also known under the name *Black-Box Optimization Benchmarking* (BBOB) referring to the BBOB workshop series (BBOB, 2009–2022) held mainly on the *Genetic and Evolutionary Computation Conference* (GECCO) conferences. COCO provides means to run an optimization algorithm on a set of benchmark optimization problems, stores the information about evaluated points, post-processes the obtained information and, finally, prepares resources to analyze the performance of the compared algorithms.

COCO provides test suites for single-objective, bi-objective, noiseless, noisy, large-scale, and mixed-integer benchmarking (Hansen et al., 2021). In this thesis, we utilize only the single-objective noiseless and noisy part of the framework, which comprises a set of 24 noiseless and 30 noisy benchmark functions of different difficulty (Hansen et al., 2009a,b). All benchmarks are defined and can be evaluated over $\mathbb{R}^D$, while the actual search domain is given as $[-5, 5]^D$. In addition, the functions are defined for a number of *instances*, independently transforming the objective-function landscape, particularly via $\mathfrak{bb}$-values scaling and translation and rotation of the input space, resulting in a set of similar but not identical objective functions. For the majority of functions, the global optimum $\mathbf{x}_{\text{opt}}$ is after the mentioned transformations located in $[-4, 4]^D$. To collect "sufficient" amount of data and make statistical analyses of results more meaningful, $N_{\text{trial}} = 15$ such instances are suggested to be used per each function to test the algorithm.

The framework delivers also postprocessing of the measured results. The core measure of an algorithm's performance is based on the algorithm's *runtime* on a single instance, i. e., the number of evaluations it needed to reach a specified target distance to the optimum $\Delta\mathfrak{bb}^*$. The *expected running time* (ERT), the expected number of function evaluations to reach a target function value for the first time is computed by dividing the sum of all evaluations before the target was reached (from successful and unsuccessful runs) by the number of runs that reached the target (Hansen et al., 2021):

$$ERT(\mathfrak{bb}_t) = \frac{\#\text{FE}s(\mathfrak{bb}_{\text{best}} \geq \mathfrak{bb}_t)}{\#\,succ}\,, \tag{65}$$

where the $\#\text{FE}s(\mathfrak{bb}_{\text{best}} \geq \mathfrak{bb}_t)$ is the number of FEs conducted in all trials, while the best function value was not smaller than $\mathfrak{bb}_t$ during the trial. The $\#\,succ$ is the number of succesful trials.

To summarize the results from a set of benchmark problems for a fixed $\mathfrak{bb}_t$, *empirical cumulative distribution function* (ECDF) of runtimes (or ECDF distributions) to reach a given single target precision valua is often used. The authors of the COCO framework favour absolute runtime distributions due to possibility of meaningful aggregation of the results from different benchmark functions, comparability accross different publications, and possibility to distinguish between easy and difficult problems.

Moreover, the COCO postprocessing defines a number of different targets to assess different difficulties of the benchmark function. For each function, 30–50 targets $\mathfrak{bb}_t = \mathfrak{bb}_{\text{opt}} + \Delta\mathfrak{bb}$ are defined, either relatively according to empirical results of a reference algorithm, or equidistantly in the logarithmic scale for $\Delta\mathfrak{bb} \in [100, 10^{-8}]$. See Figure 3 for such a ECDF graph from the COCO postprocessing.

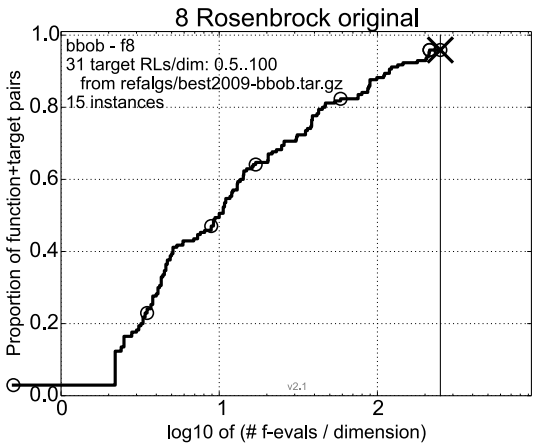


Figure 3: COCO-generated Runtime distribution (ECDF) of BOBYQA by Powell (2009) on ROSENBROCK FUNCTION f_8 in 10D from (Bajer, 2018). 31 targets per dimension are based on the best-reached $\mathbb{b}$-values of the *best2009* artificial algorithm for FE/$D \in [0.5, 100]$. The *best2009* is a dataset collected from the best-achieved results per each benchmark function and dimension from all algorithms submitted to the 2009 BBOB Workshop.

2.6.2 *Postprocessing of Convergence Plot*

Despite the statistical interpretability of the COCO ECDF graphs, in this thesis, we often interpret the performance of algorithms using median- and possibly the quartile-based convergence graphs. The aggregation of convergence graphs of one algorithm through multiple benchmarks is done on median logarithmic distances to the true optima that are linearly re-scaled to $[-8, 0]$ (Pitra et al., 2016). Such scaling is dependent only on the performance of the particular algorithms that were comprised in the comparison. That aggregation has the advantage of providing the results for each possible number of FEs. The resulting graphs are dependent on decisions about the numbers of FE used, which are always made with respect to the available evaluations budget. This is similar to the ECDF graphs from COCO postprocessing dependent on the selection of the targets to take into consideration, and, therefore, are always relative to some other parameters or choice of a reference algorithm.

The graphs using the mentioned interpretation, first presented in (Pitra et al., 2016), depict the scaled best-achieved logarithms $\Delta_f^{\log}$ of median distances Δ_f^{med} to the optimum of function f for the respective number of function evaluations per dimension FE/D. Medians Δ_f^{med} (and in some graphs also 1st and 3rd quartiles) are usually calculated from 15 independent instances for each respective algorithm, function, and dimension. The scaled logarithms of Δ_f^{med} are calculated as

$$\Delta_f^{\log} = \frac{\log_{10} \Delta_f^{\text{med}} - \Delta_f^{\text{MIN}}}{\Delta_f^{\text{MAX}} - \Delta_f^{\text{MIN}}} \log_{10} \left(1/10^{-8}\right) + \log_{10} 10^{-8} \tag{66}$$

where Δ_f^{MIN} (Δ_f^{MAX}) is the minimum (maximum) $\log_{10} \Delta_f^{\text{med}}$ found among all the compared algorithms for the particular function f and dimension D. In this thesis the range of FE/D is usually between 0 and 250.

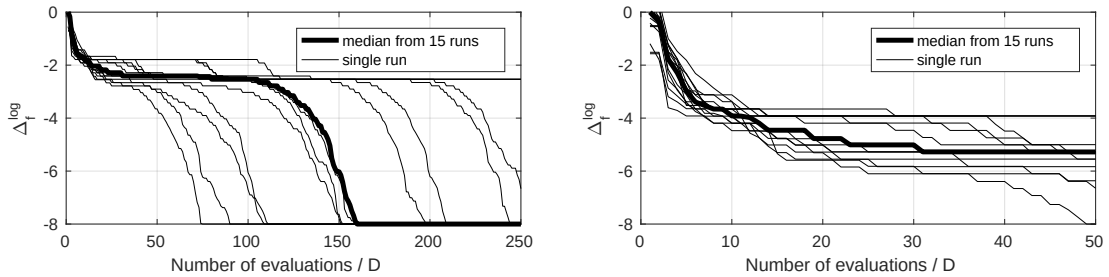


Figure 4: Convergence graphs scaled to $[-8, 0]$ from (Bajer, 2018). The graphs use the identical data as graph in Figure 3.
Left: linear scaling $[-8.00, 4.78] \to [-8, 0]$, i. e., lower limit is not scaled.
Right: data cropped to 50 FE/D, linear scaling $[-1.00, 4.78] \to [-8, 0]$.

The limits of the range $[-8, 0]$ in which $\log_{10} \Delta_f^{\text{med}}$ is scaled are motivated by the following aspects. The lower limit -8 refers to the lowest target distance 10^{-8} used in the COCO framework. Thus, reaching this target by any of the compared algorithms does not scale the lower limit (see example in Figure 4). The upper limit 0 was chosen for the reason of simplicity. The examples of scaled graphs for two different FE budgets are depicted in Figure 4.

Such scaling of median log-distances enable an aggregation of their values for the same algorithm across an arbitrary number of functions or dimensions. The results in the following chapters utilize the aggregation by average despite the possible bias by extremely different results on different functions or dimensions. As stated by Bajer (2018), we still consider this kind of aggregation as the simplest and meaningful default that provide useful information about the algorithms behaviour.

3 GOALS

After a thorough exploration of the state-of-the-art methods for surrogate-assisted evolutionary optimization of continuous black-box objectives, we identified two areas of interest with a high need for research – surrogate modeling for the CMA-ES and landscape analysis in the context of surrogate modeling for evolutionary optimization.

SURROGATE MODELING FOR THE CMA-ES Designing versions of the CMA-ES assisted by surrogate models in order to save expensive fitness evaluations and investigating their properties with the emphasis on models capable to predict the distribution of the optimized function.

LANDSCAPE ANALYSIS FOR SURROGATE MODELING Investigating the features describing the fitness landscape and their relationships to the suitability of different surrogate models in evolutionary optimization context.

4 CONTRIBUTIONS TO EVOLUTION CONTROL OF SURROGATE MODELS

This chapter partially cites the text of the articles mentioned in the following paragraph.

In this chapter, we demonstrate the advantages of the use of regression models based on Gaussian process and random forest in surrogate-assisted optimization, and the CMA-ES in particular. In Section 4.1, we present our surrogate model training procedure and methods how to select the training set. In Section 4.2, we present the *Surrogate CMA-ES* (S-CMA-ES), published in (Bajer et al., 2015; Pitra et al., 2015), algorithm employing GPs and RFs in generation-based evolution control and evaluate it together with the ^{s*}ACM-ES, another generation-based version of the CMA-ES. In Section 4.3, we propose self-adaptive version of the S-CMA-ES, published in (Repický et al., 2017), in which the number of generations using the surrogate model before retraining is adjusted depending on the performance of the last instance of the surrogate. In the next section, we present our S-CMA-ES extension, called *Doubly Trained Surrogate CMA-ES* (DTS-CMA-ES), published in (Bajer et al., 2019; Pitra et al., 2016, 2017a). The algorithm combines cheap surrogate-model predictions with the objective function evaluations. We also discuss the benefits of employing the Gaussian process uncertainty prediction, especially during the selection of points for the evaluation with a surrogate model. Our results of comparing the metric and ordinal GPs in connection with the DTS-CMA-ES published in the paper (Pitra et al., 2017b) are presented in Section 4.5. The investigation of the covariance function selection for the DTS-CMA-ES GP-based surrogate model, published in (Repický et al., 2018a,b), is reported in Section 4.6. In Section 4.7, we present our study of the DTS-CMA-ES in connection with the boosted regression forest, published in (Pitra et al., 2018a). In Section 4.8, we asses the influence of the surrogate models and their evolution control separately through experimental investigation of combinations of evolution controls and surrogate models in three surrogate-assisted versions of the CMA-ES. This research including also an analysis of these three versions was published in (Pitra et al., 2021). The investigation of the configuration of Gaussian processes serving as surrogate models in combination with artificial neural networks is proposed in Section 4.9 and was published in (Koza et al., 2021a,b; Růžička et al., 2021).

4.1 MODEL TRAINING IN SURROGATE-ASSISTED OPTIMIZATION

This section provides the insight into the training of surrogate models we utilize in all our surrogate-assisted versions of the CMA-ES: the S-CMA-ES, the DTS-CMA-ES, and their modifications. The methods are used regardless the type of currently trained surrogate model. First, we propose different methods for selection of points evaluated by the original fitness function to form a training set for surrogate model. Second, we present our procedure for training surrogate models.

4.1.1 *Training Sets*

The quality of a surrogate model is largely determined by the selection of points stored in some *archive* $\mathcal{A}$ where all original fitness evaluations are stored into its *training set* $\mathcal{T}$. We denote $N_{\max}$ the given maximum number of points in a training set. Let us list methods for *training set selection* (TSS) investigated in our research:

TSS CLOSEST, selecting $N_{\max}^{\mathcal{M}}$ points which are closest to any point in the current population (e. g., in (Bajer et al., 2019)),

TSS CLUSTER, clustering the points in the input space into $N_{\max}^{\mathcal{M}}$ clusters and taking the points nearest to clusters' centroids (e. g., in (Bajer et al., 2015)),

TSS FULL, taking all the already evaluated points, i. e., $\mathcal{T} = \mathcal{A}$ (e. g., in (Pitra et al., 2022)),

TSS KNN, selecting the union of the sets of k-nearest neighbors of all points for which the fitness should be predicted, where k is user defined (e. g., in (Kern et al., 2006)),

TSS NEAREST, selecting the union of the sets of k-nearest neighbors of all points for which the fitness should be predicted, where k is maximal such that the total number of selected points does not exceed a given maximum number of points $N_{\max}^{\mathcal{M}}$ (e. g., in (Bajer et al., 2019)), and

TSS RECENT, taking up to $N_{\max}^{\mathcal{M}}$ most recently evaluated points (e. g., in (Ulmer et al., 2003) or Loshchilov et al. (2013b)).

All of these selection methods can be restricted to an area considered relevant for the particular generation such that no point is further from a specified position than a given maximal distance $r_{\max}$. Localizing this area around the current algorithm position (around the mean $\mathbf{m}^{(g)}$ in the CMA-ES, for example) is important for local optimizers, and especially for the TSS cluster which is not biased towards the current population. The algorithms in this Chapter use the Mahalanobis distance given by the CMA-ES matrix $\sigma^2\mathbf{C}$ for selecting points into training sets, similarly to, e. g., (Kruisselbrink et al., 2010).

4.1.2 *Surrogate Model Training*

During the model training procedure in the versions of S-CMA-ES and DTS-CMA-ES, shown in general in Algorithm 5, a set of training points $\mathcal{T}$ is selected using TSS method (see Section 4.1.1). If $\mathcal{T}$ contains enough points to train the model, transformations of $\mathbf{X}_{\text{tr}}$ to the $\sigma^2\mathbf{C}$ basis and $\mathbf{y}_{\text{tr}}$ to zero mean and unit variance are calculated in Steps 4 and 5 before the model hyperparameters $\boldsymbol{\theta}$ are fitted. In case of a successful fitting procedure, the resulting model is tested for constancy on an extra generated population due to the CMA-ES restraints against stagnation.

Analogically, the predictions with such trained model $\mathcal{M}$ start with the inverse transformation of the input points in $\mathbf{X}_{\text{te}}$ into the space that was used in the model training, and the predicted f-values $\hat{\mathbf{y}}_{\text{te}}$ and variances s_{te} (if the surrogate model is able to provide them) are transformed back to the original scale. Note that the surrogate models have to save not only their settings ψ (e. g., the covariance function κ for GP models) and the corresponding trained hyperparameters $\boldsymbol{\theta}$ but also these transformations and the model's training set.

Algorithm 5 Generalized model training in S-CMA-ES and DTS-CMA-ES

Input: archive $\mathcal{A}$, TSS method, minimal number of points required to train a model $N_{\min}^{\mathcal{M}}$, model settings ψ, population size λ, CMA-ES state variables $\mathbf{m}^{(g)}$, $\sigma^{(g)}$, $\mathbf{C}^{(g)}$

1: $\mathcal{M} \leftarrow \emptyset$ {*initialize model*}
2: $\mathcal{T} = (\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \leftarrow$ select points from $\mathcal{A}$ using TSS method
3: **if** $|\mathcal{T}| \geq N_{\min}^{\mathcal{M}}$ **then**
4: $\quad \mathbf{X}_{\text{tr}} \leftarrow$ transform $\mathbf{X}_{\text{tr}}$ into the $(\sigma^{(g)})^2\mathbf{C}^{(g)}$ basis
5: $\quad \mathbf{y}_{\text{tr}} \leftarrow$ normalize $\mathbf{y}_{\text{tr}}$ to zero mean and unit variance
6: $\quad \boldsymbol{\theta} \leftarrow$ fit the hyperparameters of $\mathcal{M}$ using ψ
7: $\quad \mathbf{x}_k^{\text{test}} \leftarrow \mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)}), \quad k = 1, \ldots, \lambda$ {*create testing population*}
8: $\quad y_k^{\text{test}} \leftarrow \mathcal{M}(\mathbf{x}_k^{\text{test}}), \quad k = 1, \ldots, \lambda$ {*evaluate testing population using model with hyperparameters* $\boldsymbol{\theta}$}
9: $\quad$ **if** $\max_k(y_k^{\text{test}}) - \min_k(y_k^{\text{test}}) < \min(10^{-8}, 0.05(\max(\mathbf{y}_{\text{tr}}) - \min(\mathbf{y}_{\text{tr}}))$ **then**
10: $\quad\quad \mathcal{M}$ is considered constant $\Rightarrow$ $\mathcal{M}$ is marked as *not trained*
11: $\quad$ **end if**
12: **else**
13: $\quad \mathcal{M}$ is marked as *not trained*
14: **end if**

Output: $\mathcal{M}$ (surrogate model with hyperparameters $\boldsymbol{\theta}$)

4.2 SURROGATE CMA-ES

Our first surrogate-assisted algorithm S-CMA-ES (Bajer et al., 2015) uses the generation-based EC: once the model is trained, it is used instead of the original fitness for several consecutive generations $g_{\mathcal{M}}$ of the CMA-ES. Then, the original fitness is used for one generation, the model is re-trained also using the new points, and this procedure repeats (Algorithm 6). This EC, used for example in the s*ACM-ES, has a great advantage of simplicity and, compared to the individual-based EC, there is no need for any special selection of points into the population of the respective generation.

In order to avoid the false convergence of the algorithm in the BBOB toolbox, the model-predicted values are adapted to never be lower then the so far minimum of the original function (see the Step 12 in the pseudocode).

The main difference between the S-CMA-ES and the s*ACM-ES is in the manner how the CMA-ES is utilized. Considering the S-CMA-ES, the model prediction or training is performed within each generation of the CMA-ES. On the contrary in the s*ACM-ES, individual generations of the CMA-ES are started to optimize either original fitness, surrogate fitness, or model itself.

4.2.1 *Benchmarking S-CMA-ES using Gaussian Processes and Random Forests*

In (Bajer et al., 2015), we have compared four versions of the proposed S-CMA-ES algorithm with regular CMA-ES; Gaussian processes and random forests were used as surrogate models with two different EC settings on the noiseless part of the COCO testing framework (Hansen et al., 2012, 2009b).

Algorithm 6 S-CMA-ES

Input: original fitness function $\mathbb{bb}$, population size λ, initial step-size $\sigma^{(0)} \in \mathbb{R}_+$, initial mean $\mathbf{m}^{(0)} \in \mathbb{R}^D$, TSS method, minimal number of points required to train a model $N_{\min}^{\mathcal{M}}$, surrogate model settings ψ, the number of consecutive model-evaluated generations $g_{\mathcal{M}}$

$\mathcal{A} \leftarrow \emptyset$; $\sigma^{(0)}, \mathbf{m}^{(0)}, \mathbf{C}^{(0)} \leftarrow$ CMA-ES initialize {*initialization*}
mark $g = 0$ as original-fitness-evaluated
for generation $g = 0, 1, 2, \ldots$ until stopping conditions met **do**
  $\mathbf{x}_k \sim \mathcal{N}(\mathbf{m}^{(g)}, {\sigma^{(g)}}^2 \mathbf{C}^{(g)})$ for $k = 1, \ldots, \lambda$ {*CMA-ES sampling*}
  if g is original-fitness-evaluated **then**
    $y_k \leftarrow \mathbb{bb}(\mathbf{x}_k)$, $k = 1, \ldots, \lambda$ {*fitness evaluation*}
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{(\mathbf{x}_k, y_k)\}_{k=1}^{\lambda}$ {*archive update*}
    $\mathcal{M} \leftarrow$ trainModel$(\mathcal{A}, \text{TSS}, N_{\min}^{\mathcal{M}}, \psi, \mathbf{m}^{(g)}, \sigma^{(g)}, \mathbf{C}^{(g)})$
    mark $(g+1)$ as model-evaluated
  else
    $\hat{y}_k \leftarrow \mathcal{M}(\mathbf{x}_k)$, $k = 1, \ldots, \lambda$ {*model evaluation*}
    $\hat{y}_k \leftarrow \hat{y}_k + \max\{0, \min_{y \in \mathcal{A}} y - \min_k y_k\}$, $k = 1, \ldots, \lambda$
      {*shift values if best $\mathcal{M}$-predicted < best $\mathbb{bb}$-evaluated so far*}
    if g_m model generations have passed **then**
      mark $(g+1)$ as original-fitness-evaluated
    end if
  end if
  $\sigma^{(g+1)}, \mathbf{m}^{(g+1)}, \mathbf{C}^{(g+1)} \leftarrow$ CMA-ES update based on $\mathbf{x}_{1:\lambda}$, sorted acc. to $\mathbf{y}_{1:\lambda}$
end for

Output: $\hat{\mathbf{x}}^{\text{opt}}$ – point with the minimum achieved fitness from $\mathcal{A}$

Experimental Setup

Four S-CMA-ES algorithms were part of this study: GP1-CMAES, GP5-CMAES, RF1-CMAES and RF5-CMAES. Here, GP/RF denotes the type of the surrogate model, and the number of model-evaluated generations $g_{\mathcal{M}}$ follows. All considered S-CMA-ES versions used the TSS cluster limited by range $r_{\max} = 8$ (see Section 4.1.1). For the GP model, $\kappa_{\text{Mat}}^{5/2}$ covariance function (see Eq. (29)) with starting values $\psi = (\sigma_n^2, \ell, \sigma_f^2) = \ln(0.01, 2, 0.5)$ has been used. We have tested bagged RF comprising 100 CART trees using err_{MSE} split gain function (see Eq. (48)), each containing at least two training points in each leaf. All S-CMA-ES parameter values were chosen according to preliminary testing on several functions from the COCO framework. All the algorithms (including the CMA-ES itself) were based on the IPOP-CMA-ES version (Matlab code v. 3.61) with the following parameters: number of restarts = 4, IncPopSize = 2, $\sigma^{(0)} = \frac{8}{3}$, $\lambda = 4 + \lfloor 3 \ln D \rfloor$, and starting point $\mathbf{m}^{(0)} \sim \mathcal{U}([-4, 4]^D)$. The remainder settings were left default.

CPU Timing

In order to evaluate the CPU timing of the S-CMA-ES algorithms, we have run all the proposed algorithms on the Rosenbrock function f_8 until a maximum budget $50D$ evaluations is reached, which was far more than the required 30 seconds. The code was run on an Intel(R)

Algorithm	$2D$	$5D$	$10D$	$20D$
GP1-CMAES	0.2255	0.5713	0.1621	0.2272
GP5-CMAES	0.1870	0.2076	0.1184	0.1645
RF1-CMAES	0.2850	0.2470	0.1155	0.1395
RF5-CMAES	0.9870	0.8532	0.2315	0.1353

Table 1: The time in seconds per function evaluation for dimensions $2, 5, 10, 20$ for the S-CMA-ES algorithms.

Core(TM)2 Duo CPU E7600 @3.06GHz, 4 GB RAM with 1 processor and 2 cores. Times per function evaluation for dimensions $2, 5, 10, 20$ are registered in Table 1 for all four algorithms.

Results

Results from experiments according to Hansen et al. (2012) on all the 24 noiseless benchmark functions given in (Hansen et al., 2009b) are presented in Figures 5, 6 and 7 and in Tables 2 and 3. The ERT, used in the figures and tables, depends on a given target function value, $\mathfrak{bb}_t = \mathfrak{bb}_{opt} + \Delta\mathfrak{bb}$, and is computed over all relevant trials as the number of the original FEs executed during each trial until the best function value reached $\mathfrak{bb}_t$, summed over all trials and divided by the number of trials that actually reached $\mathfrak{bb}_t$ (Hansen et al., 2012) (see Section 2.6.1).

The most noticeable speedup of the surrogate-assisted S-CMA-ES can be observed for the "GP5" version of the S-CMA-ES (GP model, $g_m = 5$), especially in case of higher target values $\mathfrak{bb}_t$, i. e., in earlier parts of the optimization progress. The graphs in Figure 5 reveal that the GP5-CMA-ES obtained the best results on nine functions (f_2, f_5, f_7, $f_{10,11}$, f_{13-15}, f_{23}) in at least 3 out of 4 dimensionalities among the four tested S-CMA-ES versions and the original CMA-ES. This fact is also clear from the ECDFs in Figures 6 and 7 where practically all the central parts (roughly between 10 and 100FEs/D) are dominated by the GP5 model.

However, the dominant algorithms change if we consider lower (tighter) target values or later parts of the optimizations, which can be seen from the ECDF graphs. Here, the GP5-CMA-ES is always outperformed by the GP1-CMA-ES ($g_{\mathcal{M}} = 1$) and often also by other algorithms: by the original CMA-ES and, in $20D$, by the RF1-CMA-ES, too.

Random forest S-CMA-ES outperforms other algorithms only rarely in $5D$, usually in early stages of the optimization (on $f_{12,13}$, f_{16}, f_{17}, f_{22}, and f_{23}), but it is the best algorithm on 5-dimensional f_{18}. Nevertheless, the RF performance on $20D$ benchmarks is considerably more balanced and the RF models can be here considered more robust than the GP models, which start to suffer from higher dimensions (on f_{12} and f_{20}).

4.2.2 *Comparison of IPOP and BIPOP versions of S-CMA-ES*

In (Pitra et al., 2015), we have systematically compared the original CMA-ES, the surrogate-assisted S-CMA-ES using GP and RF continuous regression models with ^{s}*ACM-ES-k algorithm based on ordinal regression by Ranking SVR. These four algorithms (CMA-ES, GP-CMA-ES, RF-CMA-ES, ^{s}*ACM-ES) are tested in their IPOP version (based on the IPOP-CMA-ES by Auger and Hansen (2005)) and in the bi-population restart strategy version (based on the BIPOP-CMA-ES and its derivatives by Hansen (2009a)).

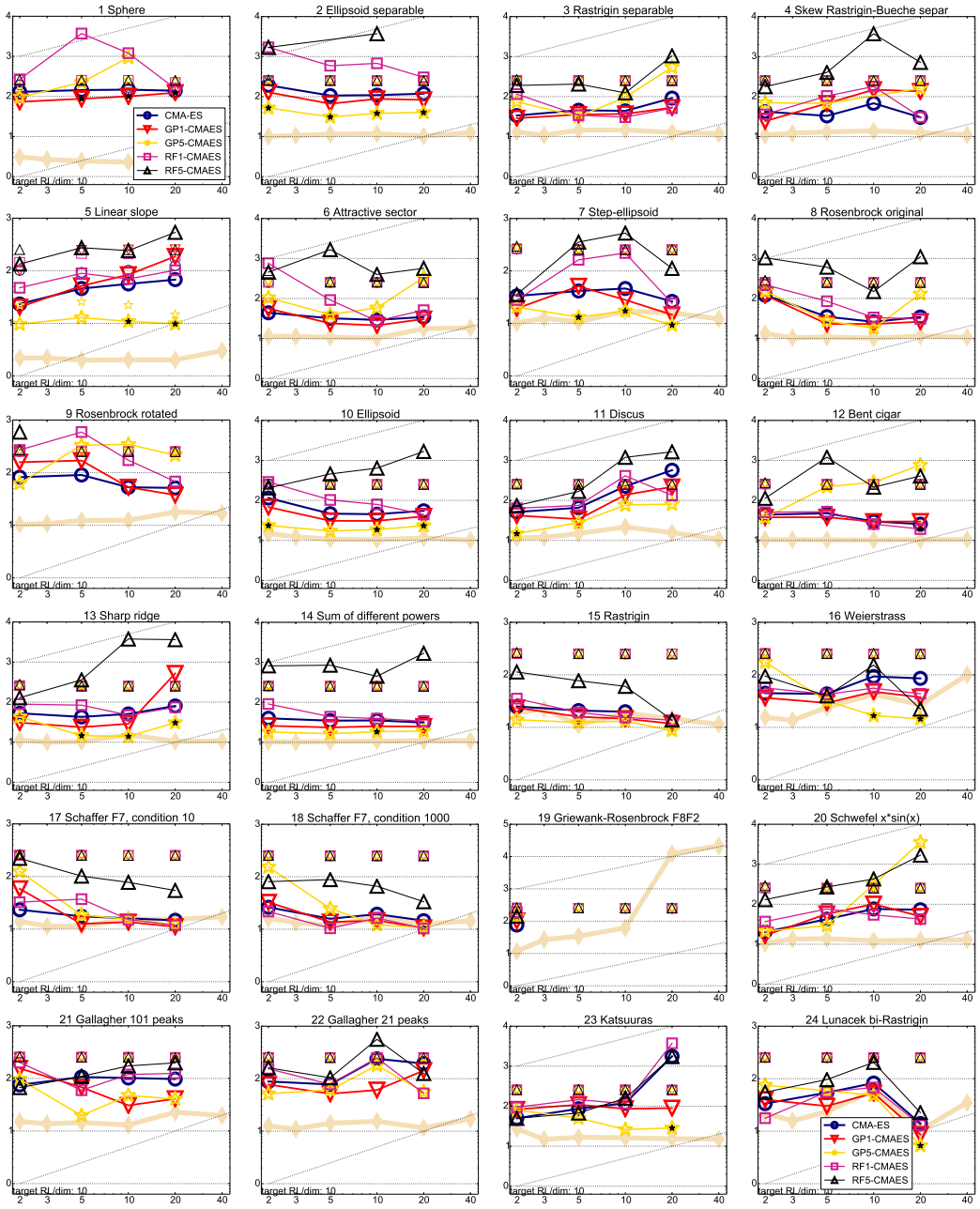


Figure 5: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value) divided by dimension versus dimension. The target function value is chosen such that the bestGECCO2009 artificial algorithm just failed to achieve an ERT of $10 \times D$. Different symbols correspond to different algorithms 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: ○:CMA-ES, ▽:GP1-CMAES, ⋆:GP5-CMAES, □:RF1-CMAES, △:RF5-CMAES

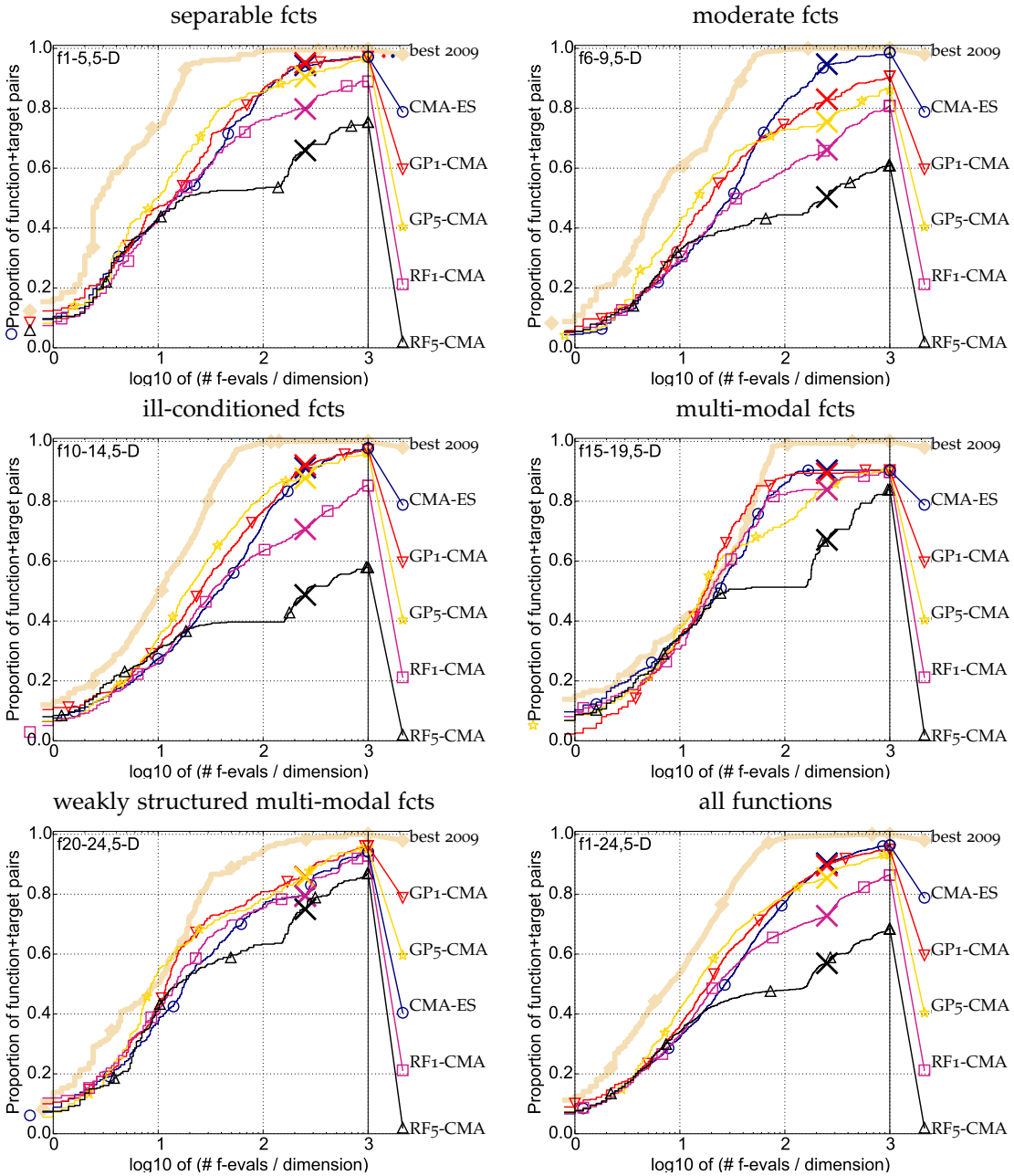


Figure 6: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for all functions and subgroups in $5D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times D$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" curve corresponds to the best ERT observed during BBOB 2009 for each selected target.

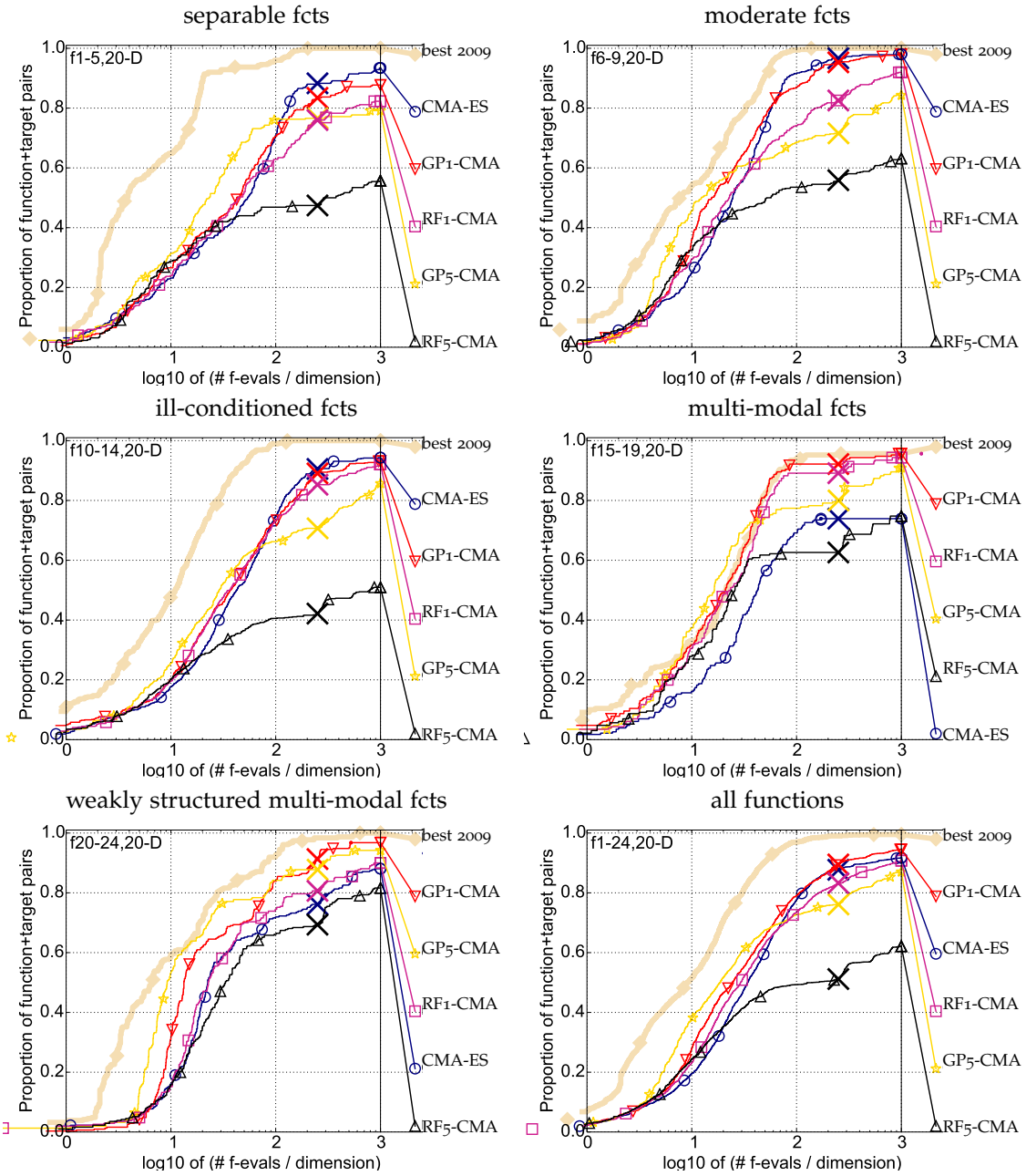


Figure 7: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for all functions and subgroups in $20D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times D$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" curve corresponds to the best ERT observed during BBOB 2009 for each selected target.

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*2.5e+1*:4.8	*1.6e+1*:7.6	*1.0e-8*:12	*1.0e-8*:12	*1.0e-8*:12	15/15
CMA-ES	2.8(4)	2.9(4)	59(7)	59(8)	59(9)	45/45
GP1-CMA	2.5(2)	2.2(2)	**36**(4)^{*2}	**36**(5)^{*2}	**36**(5)^{*2}	15/15
GP5-CMA	2.3(2)	**2.1**(1)	92(69)	92(79)	92(84)	11/15
RF1-CMA	3.7(1)	3.0(2)	1520(1624)	1520(2062)	1520(696)	1/15
RF5-CMA	**2.2**(3)	2.4(2)	∞	∞	∞ *1252*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f2	*1.6e+6*:2.9	*4.0e+5*:11	*4.0e+4*:15	*6.3e+2*:58	*1.0e-8*:95	15/15
CMA-ES	2.5(5)	**1.5**(2)	5.4(2)	9.2(5)	∞ *1258*	0/45
GP1-CMA	**2.1**(1.0)	1.9(1)	4.0(2)	5.8(4)	∞ *1258*	0/15
GP5-CMA	2.5(2)	1.7(2)	**3.2**(1)	**2.7**(1)*	**94**(191)^{*3}	2/15
RF1-CMA	2.9(3)	2.7(2)	6.8(6)	51(46)	∞ *1258*	0/15
RF5-CMA	2.1(1)	7.1(4)	25(31)	∞	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f3	*1.6e+2*:4.1	*1.0e+2*:15	*6.3e+1*:23	*2.5e+1*:73	*1.0e+1*:716	15/15
CMA-ES	3.2(2)	1.8(2)	2.6(2)	3.1(1)	**1.1**(0.6)	37/45
GP1-CMA	2.8(4)	1.5(2)	2.3(1.0)	2.5(1)	1.6(0.8)	11/15
GP5-CMA	3.0(3)	**1.3**(0.6)	**1.6**(1)	**2.2**(2)	2.6(2)	8/15
RF1-CMA	**1.9**(3)	1.6(1)	2.2(0.2)	2.4(1)	3.0(4)	6/15
RF5-CMA	3.4(3)	1.6(0.9)	4.6(10)	14(3)	6.1(6)	4/15

#FEs/D	0.5	1.2	3	10	50	#succ
f4	*2.5e+2*:2.6	*1.6e+2*:10	*1.0e+2*:19	*4.0e+1*:65	*1.6e+1*:434	15/15
CMA-ES	3.5(4)	2.4(3)	2.8(1)	**2.5**(0.9)	**1.1**(1.0)*	43/45
GP1-CMA	2.7(2)	2.4(3)	2.6(2)	5.1(10)	3.2(3)	9/15
GP5-CMA	**1.6**(1)	**1.6**(3)	3.0(1)	5.2(7)	7.5(8)	5/15
RF1-CMA	2.4(2)	1.9(2)	**2.5**(2)	7.9(10)	12(11)	3/15
RF5-CMA	2.7(3)	1.9(3)	4.9(11)	31(29)	∞ *1252*	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
CMA-ES	2.4(3)	1.9(1)	23(18)	23(15)	23(15)	45/45
GP1-CMA	**2.0**(1)	**1.5**(0.8)	26(5)	26(37)	26(36)	15/15
GP5-CMA	3.0(1)	1.7(0.1)	**6.4**(3)	**6.4**(4)	**6.4**(3)	15/15
RF1-CMA	2.3(1)	1.7(0.4)	45(47)	45(50)	45(32)	15/15
RF5-CMA	2.8(3)	1.9(1.0)	137(110)	137(185)	137(185)	10/15

#FEs/D	0.5	1.2	3	10	50	#succ
f6	*1.0e+5*:3.0	*2.5e+4*:8.4	*1.0e+2*:16	*2.5e+1*:54	*2.5e-1*:254	15/15
CMA-ES	3.9(4)	2.0(3)	3.2(4)	3.0(2)	**2.1**(0.3)^{*6}	45/45
GP1-CMA	**2.2**(2)	**1.4**(1)	**2.3**(2)	**2.2**(4)	74(46)	1/15
GP5-CMA	3.0(2)	1.8(1)	2.8(7)	3.7(8)	∞ *1260*	0/15
RF1-CMA	3.0(2)	2.0(1)	4.5(5)	8.5(4)	∞ *1258*	0/15
RF5-CMA	3.6(4)	2.4(1)	19(39)	154(36)	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f7	*1.6e+2*:4.2	*1.0e+2*:6.2	*2.5e+1*:20	*4.0e+0*:54	*1.0e+0*:324	15/15
CMA-ES	2.6(1)	3.1(4)	3.4(2)	3.8(2)	1.7(2)	43/45
GP1-CMA	**1.7**(2)	**2.0**(2)	2.1(1)	4.7(7)	1.4(1)	15/15
GP5-CMA	2.2(2)	2.5(2)	**1.9**(0.8)	**1.2**(0.7)*	**0.82**(0.7)	15/15
RF1-CMA	1.9(2)	2.6(2)	2.7(2)	15(27)	10(6)	5/15
RF5-CMA	2.5(3)	2.7(2)	5.3(9)	33(41)	17(17)	3/15

#FEs/D	0.5	1.2	3	10	50	#succ
f8	*1.0e+4*:4.6	*6.3e+3*:6.8	*1.0e+3*:18	*6.3e+1*:54	*1.6e+0*:258	15/15
CMA-ES	3.0(3)	2.8(1)	2.7(2)	3.2(5)	**4.1**(4)	37/45
GP1-CMA	2.8(4)	2.2(2)	2.3(1)	**2.1**(1)	7.2(3)	8/15
GP5-CMA	**2.1**(1)	**1.9**(1)	**1.9**(2)	2.5(1)	70(99)	1/15
RF1-CMA	2.6(2)	2.3(1)	3.1(2)	7.8(8)	36(38)	2/15
RF5-CMA	2.3(2)	2.0(2)	2.8(4)	56(31)	∞ *1252*	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
CMA-ES	**10**(3)	**8.0**(5)	**7.0**(4)	**7.3**(3)	**44**(55)	5/45
GP1-CMA	12(12)	10(13)	8.2(8)	14(36)	∞ *1258*	0/15
GP5-CMA	15(6)	14(18)	13(9)	27(24)	70(90)	1/15
RF1-CMA	38(59)	36(36)	30(48)	48(36)	∞ *1258*	0/15
RF5-CMA	106(47)	126(132)	257(486)	∞	∞ *1252*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f10	*2.5e+6*:2.9	*6.3e+5*:7.0	*2.5e+5*:17	*6.3e+3*:54	*2.5e+1*:297	15/15
CMA-ES	1.9(1)	2.0(2)	1.5(2)	4.4(2)	4.4(6)	32/45
GP1-CMA	**1.2**(1)	**0.94**(1)	**0.85**(0.8)	2.9(2)	2.0(0.9)	15/15
GP5-CMA	2.4(2)	1.6(2)	1.0(1)	**1.6**(0.7)	**0.95**(0.2)^{*3}	15/15
RF1-CMA	3.2(2)	2.3(1)	1.6(1)	10(2)	63(33)	1/15
RF5-CMA	2.7(3)	1.7(1.0)	4.5(0.8)	43(46)	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f11	*1.0e+6*:3.0	*6.3e+4*:6.2	*6.3e+2*:16	*6.3e+1*:74	*6.3e-1*:298	15/15
CMA-ES	2.0(2)	2.6(1)	4.7(3)	4.4(3)	63(123)	3/45
GP1-CMA	**1.4**(0.2)	2.4(2)	4.5(3)	2.3(2)	5.1(5)	11/15
GP5-CMA	1.5(1)	2.5(2)	**2.9**(2)	**1.9**(1.0)	**2.3**(0.3)^{*2}	14/15
RF1-CMA	2.2(1)	2.6(3)	3.9(2)	5.1(10)	∞ *1258*	0/15
RF5-CMA	2.2(3)	**2.3**(2)	3.0(2)	11(11)	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f12	*4.0e+7*:3.6	*1.6e+7*:7.6	*4.0e+6*:19	*1.6e+4*:52	*1.0e+0*:268	15/15
CMA-ES	2.2(3)	2.5(3)	2.8(1)	4.6(1)	9.3(8)	18/45
GP1-CMA	2.0(2)	2.5(2)	**1.9**(1)	**3.7**(2)	**6.2**(12)	8/15
GP5-CMA	3.1(3)	2.9(3)	3.0(2)	21(19)	8.5(7)	6/15
RF1-CMA	2.1(2)	3.2(6)	3.2(0.8)	5.1(5)	22(15)	3/15
RF5-CMA	**1.9**(2)	**1.7**(1)	5.6(10)	117(102)	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f13	*1.0e+3*:2.8	*6.3e+2*:8.4	*4.0e+2*:17	*6.3e+1*:52	*6.3e-2*:264	15/15
CMA-ES	2.9(3)	2.1(1)	2.9(2)	4.1(1)	13(10)	16/45
GP1-CMA	**2.3**(2)	**1.6**(1)	2.0(1)	2.3(0.7)	70(79)	1/15
GP5-CMA	3.2(3)	2.1(2)	**1.6**(0.8)	**1.4**(0.8)^{*2}	**11**(7)	5/15
RF1-CMA	3.2(3)	2.0(0.8)	2.4(1)	8.1(7)	69(82)	1/15
RF5-CMA	3.5(4)	2.2(0.9)	2.1(2)	35(35)	∞ *1252*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f14	*1.6e+1*:3.0	*1.0e+1*:10	*6.3e+0*:15	*2.5e-1*:53	*1.0e-5*:251	15/15
CMA-ES	3.5(4)	1.9(1)	2.1(2)	3.3(1)	**20**(15)^{*2}	11/45
GP1-CMA	3.2(5)	1.6(2)	1.9(0.9)	2.2(0.9)	∞ *1258*	0/15
GP5-CMA	4.0(3)	1.8(2)	**1.5**(0.9)	**1.6**(0.5)	∞ *1260*	0/15
RF1-CMA	3.2(2)	2.1(1)	2.8(1)	4.2(2)	∞ *1258*	0/15
RF5-CMA	**2.8**(2)	**1.2**(1)	2.0(3)	81(38)	∞ *1260*	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
CMA-ES	**2.6**(2)	**1.4**(2)	1.8(1)	1.9(1.0)	1.5(2)	43/45
GP1-CMA	4.6(5)	1.9(1)	1.9(1)	1.4(0.4)	**1.2**(1.0)	15/15
GP5-CMA	3.5(4)	1.6(1)	**1.6**(0.6)	**1.1**(0.4)	3.0(4)	11/15
RF1-CMA	5.4(3)	2.2(2)	2.4(1)	1.7(1)	1.3(0.4)	14/15
RF5-CMA	3.5(4)	1.6(0.9)	2.4(3)	7.0(8)	5.7(4)	8/15

#FEs/D	0.5	1.2	3	10	50	#succ
f16	*4.0e+1*:4.8	*2.5e+1*:16	*1.6e+1*:46	*1.0e+1*:120	*4.0e+0*:334	15/15
CMA-ES	1.4(0.9)	1.3(1)	1.7(2)	1.8(1.0)	1.9(2)	40/45
GP1-CMA	1.6(1)	1.3(2)	**0.90**(0.6)	**1.2**(2)	**1.4**(2)	13/15
GP5-CMA	2.0(2)	1.5(1)	2.7(4)	1.3(0.2)	1.9(3)	13/15
RF1-CMA	**1.3**(1)	**0.90**(0.8)	1.3(0.6)	1.8(1)	2.1(3)	11/15
RF5-CMA	2.5(1)	1.3(1)	1.1(1)	1.7(5)	3.2(3)	9/15

#FEs/D	0.5	1.2	3	10	50	#succ
f17	*1.0e+1*:5.2	*6.3e+0*:26	*4.0e+0*:57	*2.5e+0*:110	*6.3e-1*:412	15/15
CMA-ES	3.2(3)	1.7(1)	1.5(1.0)	1.2(1)	0.72(1)	44/45
GP1-CMA	4.5(4)	**1.6**(0.8)	**1.1**(0.5)	**0.78**(0.3)	**0.45**(0.2)	15/15
GP5-CMA	3.6(5)	1.6(2)	1.6(1)	1.8(4)	2.5(2)	10/15
RF1-CMA	**3.0**(2)	2.2(2)	3.3(6)	3.9(9)	4.1(9)	7/15
RF5-CMA	4.8(7)	8.6(15)	9.0(10)	10(11)	22(17)	2/15

#FEs/D	0.5	1.2	3	10	50	#succ
f18	*6.3e+1*:3.4	*4.0e+1*:7.2	*2.5e+1*:20	*1.6e+1*:58	*1.6e+0*:318	15/15
CMA-ES	2.7(4)	2.6(5)	2.0(1)	1.4(0.5)	**1.1**(0.5)	44/45
GP1-CMA	2.4(5)	3.6(3)	2.1(2)	1.2(0.6)	1.2(2)	14/15
GP5-CMA	1.7(0.3)	3.0(3)	5.2(6)	2.1(6)	5.8(8)	7/15
RF1-CMA	**1.5**(3)	**1.7**(2)	**1.3**(0.8)	**0.89**(0.7)	2.2(3)	11/15
RF5-CMA	2.2(3)	3.0(2)	9.5(10)	7.6(8)	29(44)	2/15

#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
CMA-ES	∞	∞	∞	∞	∞ *1258*	0/45
GP1-CMA	∞	∞	∞	∞	∞ *1260*	0/15
GP5-CMA	∞	∞	∞	∞	∞ *1262*	0/15
RF1-CMA	∞	∞	∞	∞	∞ *1258*	0/15
RF5-CMA	∞	∞	∞	∞	∞ *1262*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f20	*6.3e+3*:5.1	*4.0e+3*:8.4	*4.0e+1*:15	*2.5e+0*:69	*1.0e+0*:851	15/15
CMA-ES	2.5(2)	2.1(3)	3.7(3)	3.1(2)	13(15)	5/45
GP1-CMA	**1.9**(2)	1.8(1)	3.0(2)	4.2(3)	**11**(12)	2/15
GP5-CMA	2.5(2)	1.8(2)	**2.1**(0.5)	**2.1**(2)	∞ *1260*	0/15
RF1-CMA	2.9(4)	2.4(2)	3.9(2)	5.6(5)	∞ *1258*	0/15
RF5-CMA	2.5(3)	**1.8**(1)	25(29)	20(24)	∞ *1260*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f21	*4.0e+1*:3.9	*2.5e+1*:11	*1.6e+1*:31	*6.3e+0*:73	*1.6e+0*:347	5/5
CMA-ES	2.2(4)	1.7(2)	1.5(2)	7.3(18)	10(19)	13/45
GP1-CMA	**1.1**(1)	**1.4**(2)	0.99(0.8)	4.7(5)	4.9(11)	7/15
GP5-CMA	2.5(3)	1.7(1)	**0.89**(0.5)	**1.4**(3)	**4.7**(3)	8/15
RF1-CMA	2.0(2)	1.8(2)	1.4(2)	4.1(1)	4.8(5)	7/15
RF5-CMA	1.8(0.9)	2.6(2)	2.9(7)	7.5(8)	10(10)	5/15

#FEs/D	0.5	1.2	3	10	50	#succ
f22	*6.3e+1*:3.6	*4.0e+1*:15	*2.5e+1*:32	*1.0e+1*:71	*1.6e+0*:341	5/5
CMA-ES	3.4(3)	2.3(1)	1.7(1)	5.5(6)	8.9(15)	14/45
GP1-CMA	2.6(3)	**1.4**(1)	**1.2**(0.8)	**3.6**(9)	10(12)	4/15
GP5-CMA	2.9(3)	7.5(7)	4.6(6)	4.3(3)	11(8)	4/15
RF1-CMA	3.1(2)	2.0(2)	1.3(0.9)	5.5(0.5)	**3.2**(4)	9/15
RF5-CMA	**2.1**(2)	4.7(1.0)	2.9(13)	7.3(8)	24(39)	2/15

#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
CMA-ES	2.5(2)	2.0(3)	3.1(3)	5.2(9)	26(55)	4/45
GP1-CMA	1.9(2)	2.7(3)	3.2(4)	6.5(4)	4.9(4)	6/15
GP5-CMA	2.4(1)	2.0(1)	**2.4**(2)	**3.0**(5)	**2.2**(1)	11/15
RF1-CMA	**1.8**(2)	2.0(4)	3.0(3)	8.7(10)	∞ *1260*	0/15
RF5-CMA	2.4(2)	**1.5**(0.9)	3.3(3)	4.0(4)	∞ *1288*	0/15

#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
CMA-ES	2.3(2)	2.4(1)	2.4(1)	2.3(1)	1.6(2)	33/45
GP1-CMA	1.7(1.0)	1.7(0.6)	1.7(1)	**1.3**(0.7)	**1.1**(0.5)	13/15
GP5-CMA	**1.5**(0.9)	**1.1**(0.4)	**1.1**(0.5)	2.5(6)	1.6(2)	9/15
RF1-CMA	1.9(2)	2.4(1)	2.4(1)	2.4(6)	1.9(2)	9/15
RF5-CMA	1.8(1)	3.6(11)	3.6(6)	4.1(7)	2.9(2)	7/15

Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 5. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear for each algorithm and run-length based target, the corresponding best ERT (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 by the number of instances.

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*6.3e+1*:24	*4.0e+1*:42	*1.0e-8*:43	*1.0e-8*:43	*1.0e-8*:43	15/15
CMA-ES	4.9(1)	4.5(1)	64(3)	64(3)	64(4)	45/45
GP1-CMA	3.9(1)	3.1(0.6)	**58**(5)*2	**58**(9)*2	**58**(6)*2	15/15
GP5-CMA	**2.9**(0.3)	**2.0**(0.2)*	∞	∞	∞ *5034*	0/15
RF1-CMA	3.9(2)	3.5(0.5)	73(15)	73(19)	73(9)	15/15
RF5-CMA	3.7(2)	3.0(1)	∞	∞	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f2	*4.0e+6*:29	*2.5e+6*:42	*1.0e+5*:65	*1.0e+4*:207	*1.0e-8*:412	15/15
CMA-ES	1.0(0.9)	1.3(0.9)	14(5)	11(2)	∞ *5006*	0/45
GP1-CMA	1.5(1)	1.6(1)	8.9(2)	8.0(3)	∞ *5006*	0/15
GP5-CMA	**0.90**(0.7)	1.3(1.0)	**5.4**(2)*2	**3.9**(0.7)*3	∞ *5006*	0/15
RF1-CMA	1.1(0.6)	**1.2**(0.6)	11(5)	29(12)	∞ *5006*	0/15
RF5-CMA	1.2(0.7)	1.3(1.0)	184(350)	∞	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f3	*6.3e+2*:33	*4.0e+2*:44	*1.6e+2*:109	*1.0e+2*:255	*2.5e+1*:3277	15/15
CMA-ES	2.0(1)	4.0(1)	7.5(3)	7.2(2)	**7.8**(6)	8/45
GP1-CMA	2.3(0.8)	3.2(1)	**5.9**(2)	4.2(1.0)	22(30)	1/15
GP5-CMA	1.9(0.7)	2.9(0.6)	15(15)	43(50)	∞ *5034*	0/15
RF1-CMA	2.0(1)	3.7(1)	6.4(1)	**4.0**(0.5)	22(23)	1/15
RF5-CMA	**1.7**(0.4)	**2.6**(0.6)	18(47)	82(104)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f4	*6.3e+2*:22	*4.0e+2*:91	*2.5e+2*:250	*1.6e+2*:332	*6.3e+1*:1927	15/15
CMA-ES	7.7(3)	3.4(0.9)	**2.4**(0.5)	**4.1**(0.8)	**2.4**(2)*6	30/45
GP1-CMA	9.2(4)	4.5(4)	11(11)	42(26)	39(34)	1/15
GP5-CMA	**7.5**(3)	5.5(5)	13(15)	215(227)	∞ *5022*	0/15
RF1-CMA	8.1(4)	**3.4**(1)	2.5(0.6)	7.7(8)	∞ *5006*	0/15
RF5-CMA	8.9(5)	13(2)	57(120)	∞	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f5	*2.5e+2*:19	*1.6e+2*:34	*1.0e-8*:41	*1.0e-8*:41	*1.0e-8*:41	15/15
CMA-ES	1.6(2)	1.9(1)	33(13)	33(13)	33(14)	45/45
GP1-CMA	**1.6**(1)	1.9(0.3)	92(239)	92(66)	92(78)	11/15
GP5-CMA	1.7(1)	**1.7**(0.0)	**4.8**(1)*4	**4.8**(2)*4	**4.8**(1)*4	15/15
RF1-CMA	1.8(1)	2.3(1)	50(20)	50(29)	50(34)	15/15
RF5-CMA	2.0(0.9)	2.0(0.6)	265(232)	265(308)	265(294)	6/15
#FEs/D	0.5	1.2	3	10	50	#succ
f6	*2.5e+5*:16	*6.3e+4*:43	*1.6e+4*:62	*1.6e+2*:353	*1.6e+1*:1078	15/15
CMA-ES	3.0(2)	2.4(2)	2.7(1)	2.0(0.5)	**1.5**(0.2)*4	45/45
GP1-CMA	2.8(1)	2.0(0.9)	2.0(0.6)	**1.7**(0.7)	4.2(5)	11/15
GP5-CMA	2.6(1)	1.6(0.5)	**1.5**(0.3)	19(21)	∞ *5024*	0/15
RF1-CMA	2.7(1)	2.1(0.8)	2.3(1)	2.9(1)	66(91)	1/15
RF5-CMA	**2.1**(0.9)	**1.6**(1)	1.8(0.3)	32(56)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f7	*1.0e+3*:11	*4.0e+2*:39	*2.5e+2*:74	*6.3e+1*:319	*1.0e+1*:1351	15/15
CMA-ES	**1.7**(2)	2.4(0.6)	2.6(0.7)	1.7(0.5)	**1.1**(0.3)	45/45
GP1-CMA	2.1(2)	2.2(1)	1.7(0.4)	0.92(0.3)	3.0(3)	10/15
GP5-CMA	2.4(2)	**1.8**(0.4)	**1.3**(0.3)	**0.58**(0.1)*2	1.6(1)	14/15
RF1-CMA	2.0(2)	2.6(1)	2.3(0.9)	1.5(0.9)	54(123)	1/15
RF5-CMA	1.8(2)	1.9(0.9)	1.8(0.2)	7.0(7)	∞ *5034*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f8	*4.0e+4*:19	*2.5e+4*:35	*4.0e+3*:67	*2.5e+2*:231	*1.6e+1*:1470	15/15
CMA-ES	6.5(2)	4.7(2)	4.5(0.7)	3.0(0.7)	3.6(3)	31/45
GP1-CMA	5.5(1)	3.4(1)	3.1(0.6)	**2.3**(1)	**3.0**(1)	12/15
GP5-CMA	**4.4**(0.6)	**2.6**(0.2)*	**2.2**(0.5)*2	11(11)	8.2(5)	5/15
RF1-CMA	5.8(1)	4.0(0.7)	4.0(1)	2.7(1)	25(41)	2/15
RF5-CMA	5.8(2)	3.7(2)	4.2(2)	96(158)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f9	*1.0e+2*:357	*6.3e+1*:560	*4.0e+1*:684	*2.5e+1*:756	*1.0e+1*:1716	15/15
CMA-ES	2.9(0.5)	2.5(3)	2.2(2)	2.2(2)	∞ *5006*	0/45
GP1-CMA	**2.1**(0.9)	**1.6**(0.4)	**1.4**(0.4)	**1.5**(0.2)	∞ *5006*	0/15
GP5-CMA	12(15)	12(32)	13(30)	12(17)	∞ *5020*	0/15
RF1-CMA	3.8(1)	5.8(7)	5.3(4)	5.8(9)	∞ *5006*	0/15
RF5-CMA	∞	∞	∞	∞	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f10	*1.6e+6*:15	*1.0e+6*:27	*4.0e+5*:70	*6.3e+4*:231	*4.0e+3*:1015	15/15
CMA-ES	5.6(3)	5.5(4)	5.1(1)	4.8(0.8)	3.2(1)	45/45
GP1-CMA	5.4(4)	5.4(3)	4.2(1)	3.5(1)	2.3(0.6)	15/15
GP5-CMA	**4.8**(3)	**3.6**(1)	**3.1**(0.5)	**2.0**(0.5)*2	**1.1**(0.4)*3	15/15
RF1-CMA	5.9(4)	5.3(3)	4.1(1)	3.9(2)	74(89)	1/15
RF5-CMA	5.1(3)	5.2(3)	10(19)	145(87)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f11	*4.0e+4*:11	*2.5e+3*:27	*1.6e+2*:313	*1.0e+2*:481	*1.0e+1*:1002	15/15
CMA-ES	1.7(1)	2.3(3)	36(47)	151(143)	∞ *5006*	0/45
GP1-CMA	**1.3**(1)	**1.8**(2)	14(6)	15(13)	∞ *5006*	0/15
GP5-CMA	1.8(1)	2.3(1)	**5.2**(4)	17(17)	∞ *5008*	0/15
RF1-CMA	1.5(2)	3.2(3)	8.6(0.7)	**12**(6)	∞ *5006*	0/15
RF5-CMA	1.9(2)	2.8(2)	105(116)	146(276)	∞ *5008*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f12	*1.0e+8*:23	*6.3e+7*:39	*2.5e+7*:76	*4.0e+6*:209	*1.0e+1*:1042	15/15
CMA-ES	5.5(2)	4.9(1.0)	3.8(0.6)	2.4(0.5)	3.4(2)	35/45
GP1-CMA	4.7(3)	**3.9**(3)	3.7(0.8)	2.8(0.9)	**2.4**(2)	13/15
GP5-CMA	18(6)	22(39)	38(45)	74(66)	21(26)	3/15
RF1-CMA	**3.9**(2)	3.9(1)	**3.1**(0.6)	**1.8**(0.3)*2	3.0(1)	12/15
RF5-CMA	5.2(3)	4.6(3)	9.4(3)	39(30)	∞ *5006*	0/15

#FEs/D	0.5	1.2	3	10	50	#succ
f13	*1.6e+3*:28	*1.0e+3*:64	*6.3e+2*:79	*4.0e+1*:211	*2.5e+0*:1724	15/15
CMA-ES	3.5(2)	4.0(1)	5.3(1)	7.6(5)	6.4(9)	16/45
GP1-CMA	2.5(0.8)	2.4(0.3)	2.9(0.5)	49(86)	42(89)	1/15
GP5-CMA	**2.4**(0.6)	**1.5**(0.3)*4	**1.6**(0.2)*4	**2.9**(0.8)*2	**4.5**(6)	7/15
RF1-CMA	3.2(0.8)	3.0(0.9)	3.9(0.8)	7.1(4)	7.3(9)	5/15
RF5-CMA	3.4(1)	3.0(0.6)	4.2(1)	343(208)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f14	*2.5e+1*:15	*1.6e+1*:42	*1.0e+1*:75	*1.6e+0*:219	*6.3e-4*:1106	15/15
CMA-ES	9.1(3)	4.9(2)	4.0(1)	2.9(0.4)	**4.0**(1)	44/45
GP1-CMA	7.9(3)	3.9(2)	3.0(2)	2.3(0.9)	4.6(1)	13/15
GP5-CMA	**5.7**(1)	**2.6**(0.6)	**2.1**(0.3)	**1.7**(0.5)	67(59)	1/15
RF1-CMA	7.1(4)	4.4(0.5)	3.5(0.8)	3.1(0.9)	33(25)	2/15
RF5-CMA	6.5(4)	3.9(2)	3.7(2)	153(172)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f15	*6.3e+2*:15	*4.0e+2*:67	*2.5e+2*:292	*1.6e+2*:846	*1.0e+2*:1671	15/15
CMA-ES	**3.6**(3)	1.9(1)	0.81(0.3)	**0.78**(0.3)	**0.64**(0.2)	15/15
GP1-CMA	3.7(2)	**1.5**(0.3)	**0.60**(0.4)	2.2(5)	3.9(5)	7/15
GP5-CMA	4.5(3)	2.2(0.7)	0.93(0.2)	0.92(0.1)	0.70(0.2)	15/15
RF1-CMA	3.6(2)	1.9(0.8)	0.95(0.4)	1.7(0.4)	6.5(9)	5/15
#FEs/D	0.5	1.2	3	10	50	#succ
f16	*4.0e+1*:26	*2.5e+1*:127	*1.6e+1*:540	*1.6e+1*:540	*1.0e+1*:1384	15/15
CMA-ES	3.3(3)	10(5)	3.2(0.9)	3.2(1)	1.4(0.4)	45/45
GP1-CMA	3.3(3)	4.3(2)	1.4(0.3)	1.4(0.3)	0.90(1.0)	14/15
GP5-CMA	3.4(2)	**1.6**(0.4)	**0.54**(0.1)*2	**0.54**(0.2)*2	**0.57**(0.4)	15/15
RF1-CMA	**2.8**(4)	4.6(2)	1.6(0.5)	1.6(0.5)	0.79(0.2)	15/15
RF5-CMA	4.2(2)	2.2(1)	0.83(0.2)	0.83(0.2)	1.0(1)	13/15
#FEs/D	0.5	1.2	3	10	50	#succ
f17	*1.6e+1*:11	*1.0e+1*:63	*6.3e+0*:305	*4.0e+0*:468	*1.0e+0*:1030	15/15
CMA-ES	3.3(2)	2.2(1.0)	0.96(0.6)	1.0(0.5)	**1.0**(0.2)	45/45
GP1-CMA	**2.4**(3)	**1.4**(0.6)	**0.73**(0.2)	**0.79**(0.2)	3.4(5)	10/15
GP5-CMA	3.3(3)	1.6(0.7)	0.79(0.4)	0.87(0.3)	11(15)	5/15
RF1-CMA	3.1(3)	1.9(1)	0.78(0.3)	0.83(0.3)	4.2(6)	9/15
RF5-CMA	4.0(0.7)	2.7(1)	3.6(0.6)	17(16)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f18	*4.0e+1*:116	*2.5e+1*:252	*1.6e+1*:430	*1.0e+1*:621	*4.0e+0*:1090	15/15
CMA-ES	1.1(0.8)	1.2(0.2)	1.1(0.3)	1.1(0.5)	**1.2**(1)	44/45
GP1-CMA	**0.81**(0.4)	0.84(0.3)	**0.85**(0.3)	**0.93**(0.2)	5.2(7)	8/15
GP5-CMA	0.95(0.9)	0.85(0.6)	1.7(4)	2.8(4)	19(22)	3/15
RF1-CMA	0.87(0.4)	**0.82**(0.2)	0.87(0.3)	1.0(0.3)	10(15)	5/15
RF5-CMA	1.3(0.3)	2.7(11)	5.4(0.6)	53(40)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f19	*1.6e-1*:2.5e5	*1.0e-1*:3.4e5	*6.3e-2*:3.4e5	*4.0e-2*:3.4e5	*2.5e-2*:3.4e5	3/15
CMA-ES	∞	∞	∞	∞	∞ *5008*	0/45
GP1-CMA	∞	∞	∞	∞	∞ *5006*	0/15
GP5-CMA	∞	∞	∞	∞	∞ *5020*	0/15
RF1-CMA	∞	∞	∞	∞	∞ *5008*	0/15
RF5-CMA	∞	∞	∞	∞	∞ *5034*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f20	*1.6e+4*:38	*1.0e+4*:42	*2.5e+2*:62	*2.5e+0*:250	*1.6e+0*:2536	15/15
CMA-ES	3.3(1)	4.0(1)	5.9(1)	5.8(1)	11(7)	7/45
GP1-CMA	2.8(0.7)	3.2(1)	3.6(0.6)	4.0(0.6)	∞ *5006*	0/15
GP5-CMA	**2.3**(0.7)	**2.2**(0.6)*	**2.5**(0.4)*3	284(483)	∞ *5022*	0/15
RF1-CMA	3.5(0.6)	3.9(0.5)	5.2(1)	**3.4**(0.6)	**6.1**(5)	4/15
RF5-CMA	2.6(0.6)	2.9(0.8)	6.6(3)	134(179)	∞ *5006*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f21	*6.3e+1*:36	*4.0e+1*:77	*4.0e+1*:77	*1.6e+1*:456	*4.0e+0*:1094	15/15
CMA-ES	5.9(3)	7.5(19)	7.5(2)	4.3(8)	8.8(7)	17/45
GP1-CMA	3.8(0.9)	6.3(1)	6.3(0.9)	1.8(5)	**3.4**(6)	10/15
GP5-CMA	**3.2**(0.8)	**2.9**(0.7)	**2.9**(0.6)	**1.8**(2)	7.7(4)	6/15
RF1-CMA	5.2(2)	3.7(1)	3.7(0.9)	5.5(11)	8.0(4)	6/15
RF5-CMA	7.6(2)	6.1(3)	6.1(5)	8.7(11)	14(11)	4/15
#FEs/D	0.5	1.2	3	10	50	#succ
f22	*6.3e+1*:45	*4.0e+1*:68	*4.0e+1*:68	*1.6e+1*:231	*6.3e+0*:1219	15/15
CMA-ES	4.8(2)	10(17)	10(35)	17(30)	9.2(11)	15/45
GP1-CMA	12(29)	17(17)	17(35)	12(37)	2.7(6)	10/15
GP5-CMA	**2.5**(1)*	**3.6**(12)*2	**3.6**(7)*2	4.8(7)	**2.0**(4)	11/15
RF1-CMA	4.6(2)	5.2(4)	5.2(3)	**4.6**(2)	3.0(5)	10/15
RF5-CMA	5.8(2)	11(4)	11(4)	11(1)	5.7(7)	7/15
#FEs/D	0.5	1.2	3	10	50	#succ
f23	*6.3e+0*:29	*4.0e+0*:118	*2.5e+0*:306	*2.5e+0*:306	*1.0e+0*:1614	15/15
CMA-ES	1.4(0.8)	6.9(7)	115(150)	115(217)	∞ *5020*	0/45
GP1-CMA	2.1(2)	6.4(6)	5.8(2)	5.8(0.6)	1.6(0.7)	14/15
GP5-CMA	2.2(3)	**2.9**(1)	**1.9**(0.2)*2	**1.9**(2)*2	**0.84**(0.8)	15/15
RF1-CMA	**1.3**(1)	3.4(3)	244(267)	244(197)	∞ *5010*	0/15
RF5-CMA	1.8(4)	4.5(4)	113(109)	113(122)	∞ *5086*	0/15
#FEs/D	0.5	1.2	3	10	50	#succ
f24	*2.5e+1*:208	*1.6e+1*:918	*1.0e+2*:6628	*6.3e+1*:9885	*4.0e+1*:31629	15/15
CMA-ES	1.3(0.2)	1.7(0.9)	3.6(2)	3.2(3)	3.6(3)	2/45
GP1-CMA	0.83(0.2)	**0.84**(0.6)	**0.20**(0.0)	**0.24**(0.0)	**0.38**(0.6)	5/15
GP5-CMA	**0.51**(0.1)*2	0.86(3)	0.37(0.4)	0.32(0.4)	1.1(0.9)	2/15
RF1-CMA	1.3(0.2)	2.0(2)	0.50(0.4)	0.52(0.4)	∞ *5006*	0/7
RF5-CMA	2.2(1)	4.1(5)	11(13)	∞	∞ *5034*	0/15

Table 3: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 20. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear for each algorithm and run-length based target, the corresponding best ERT (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 by the number of instances.

Experimental setup

The experimental evaluation was performed through the noiseless part of the COCO framework (Hansen et al., 2012, 2009b) in dimensions 2, 5, 10, and 20 in 15 different instances, meaning that 1440 optimization runs were called for each of the eight considered algorithms. The following paragraphs summarize the parameters of the algorithms.

CMA-ES. The original CMA-ES was employed in its IPOP-CMA-ES version (Matlab code v. 3.61) using settings identical to Bajer et al. (2015) described in Section 4.2.1.

^{s*}ACM-ES. We have used Loshchilov's GECCO 2013 Matlab code `xacmes.m` (Loshchilov et al., 2013a) in its ^{s*}ACM-ES version, setting the parameters CMAactive = 1, newRestartRules = 0 and withSurr = 1, modelType = 1, withModelEnsembles = 0, withModelOptimization = 1, hyper_lambda = 20, $\lambda_{\text{Mult}} = 1$, $\mu_{\text{Mult}} = 1$ and $\Lambda_{\text{minIter}} = 4$.

S-CMA-ES: GP5-CMA-ES AND RF5-CMA-ES. The number after the GP/RF in the names of the algorithms denotes the number of model-evaluated generations $g_{\mathcal{M}}$, which are evaluated by the model in row. The remaining parameters were used identical to (Bajer et al., 2015) described in Section 4.2.1.

BIPOP VERSION OF THE ALGORITHMS. The bi-population versions BIPOP-CMA-ES and BIPOP-^{s*}ACM-ES use the same Loshchilov's Matlab code `xacmes.m` with the parameter BIPOP = 1. The BIPOP-GP5-CMA-ES and BIPOP-RF5-CMA-ES algorithms are constructed in the same manner as the S-CMA-ES was transformed from the CMA-ES – by integration of the Algorithm 6 into every generation of the BIPOP-CMA-ES.

Results

The performance of the algorithms is compared in the graphs placed in Figures 8–10 (see ECDF graphs in Section 2.6.1). As we can see in Figure 8, the 24 functions can be roughly divided into two groups according to the algorithm which performed the best (at least in $10D$ and $20D$). The first group of functions where the CMA-ES performed best consists of functions f_1, f_3, f_4, f_6, and f_{20} while on functions f_2, f_5, f_7, $f_{10,11}$, f_{13-16}, f_{18}, f_{21}, and $f_{23,24}$, GP5-CMA-ES is usually better. The usage of the BIPOP versions generally leads to no improvement or even to performance decrease.

We can see that our GP5-CMA-ES usually outperforms the other algorithms when we consider the evaluations budget FEs $\leq 10^{1.5}D$, i. e., FEs ≤ 150 for $5D$ and FEs ≤ 600 for $20D$. However, as the number of the considered original evaluations rises, the original CMA-ES or the ^{s*}ACM-ES usually performs better. This fact can be summarized that our GP5-CMA-ES is more convenient for the applications where a very low number of function evaluations is available, such as in (Baerns and Holeňa, 2009).

4.2.3 *Conclusion*

This section investigated two surrogate models based on Gaussian processes and random forests, which have partly similar properties (e. g., they inherently provide estimation of the prediction error). The two models were used as surrogate models for the CMA-ES, resulting

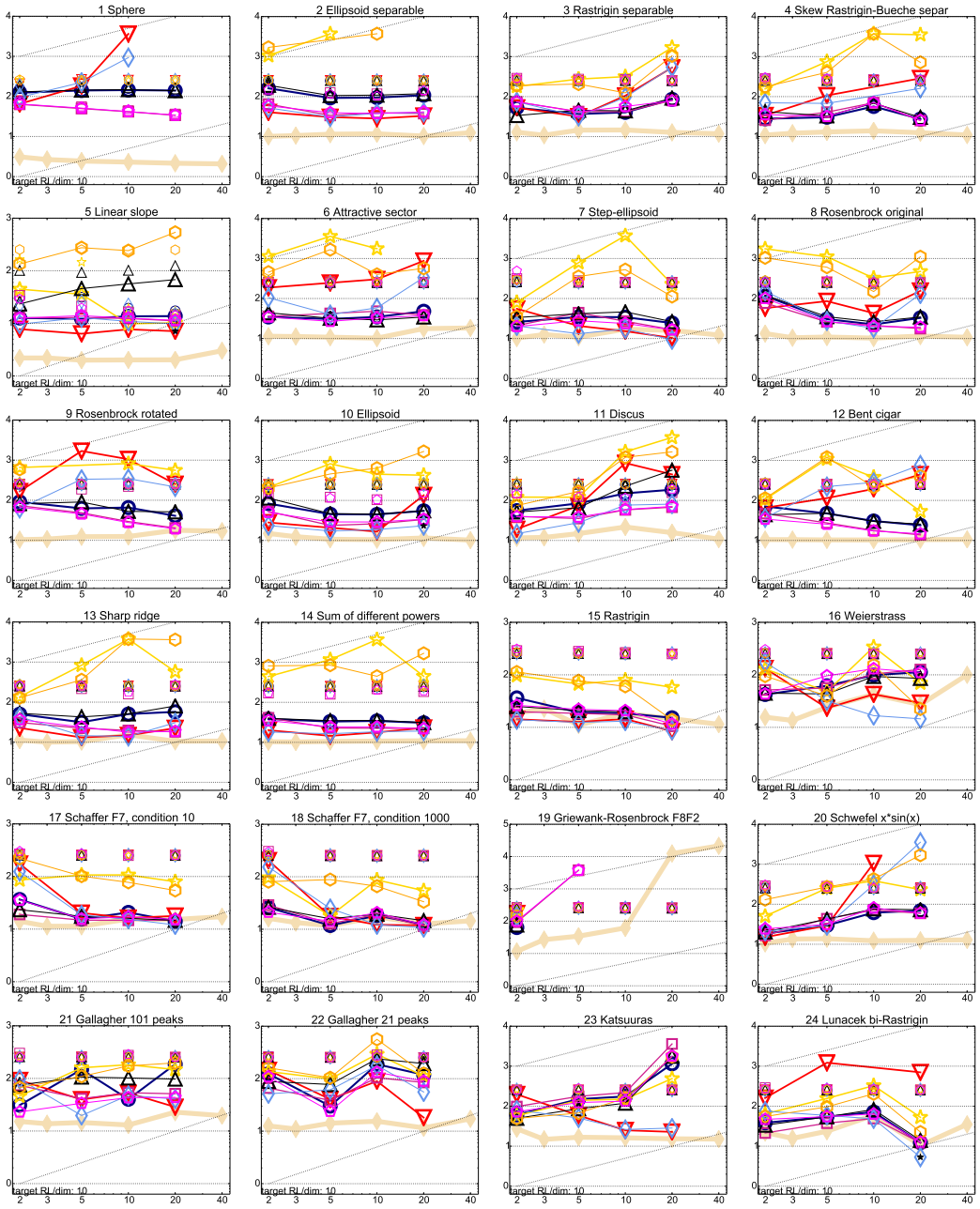


Figure 8: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value) divided by dimension versus dimension. The target function value is chosen such that the bestGECCO2009 artificial algorithm just failed to achieve an ERT of $10 \times D$. Different symbols correspond to different algorithms 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: ○:CMA-ES, ▽:GP1-CMAES, ⋆:GP5-CMAES, □:RF1-CMAES, △:RF5-CMAES, ♢:GP5-CMAES, ⬡:RF5-CMAES, ⬠:saACMES

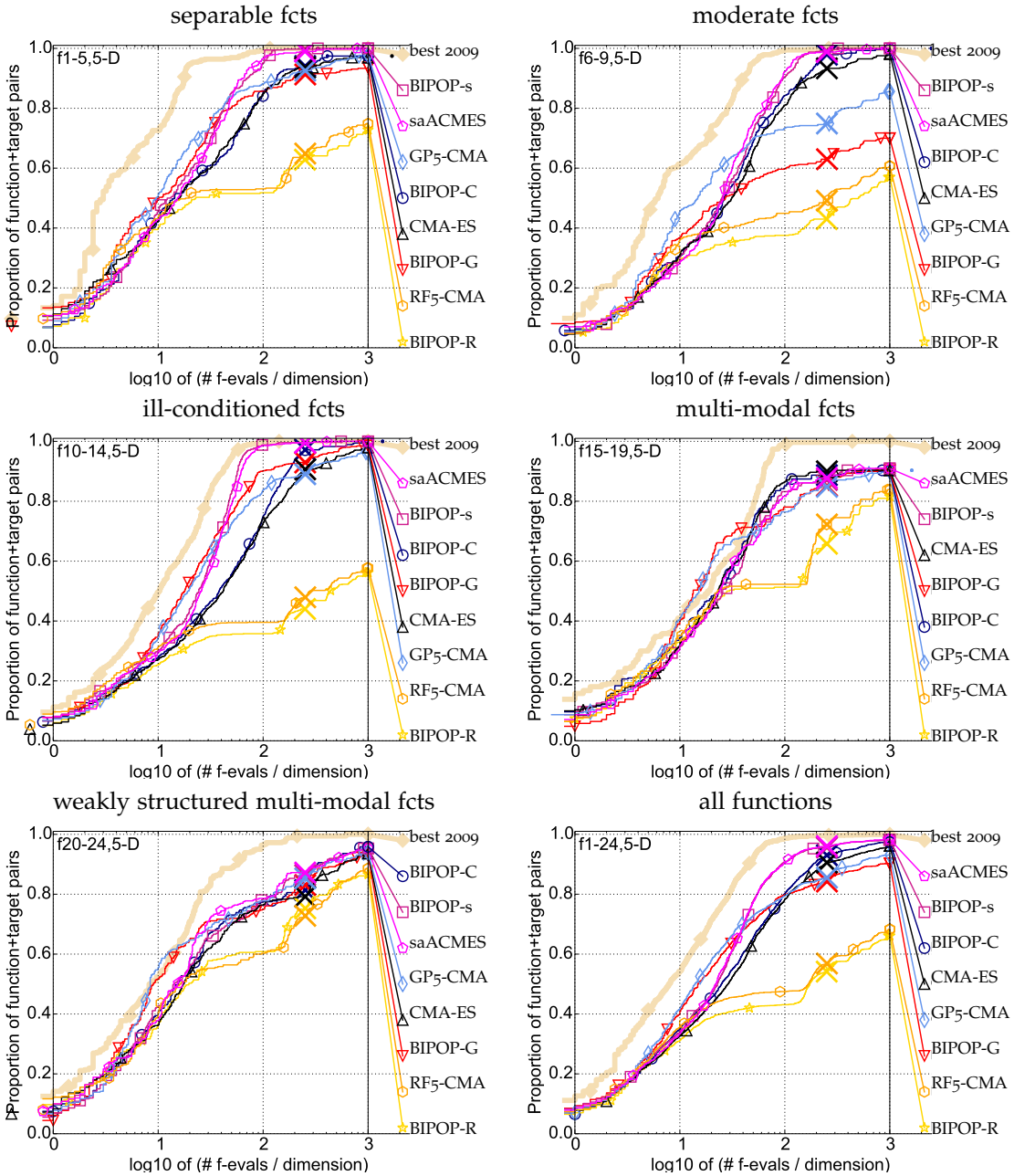


Figure 9: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for all functions and subgroups in $5D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times D$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" curve corresponds to the best ERT observed during BBOB 2009 for each selected target.

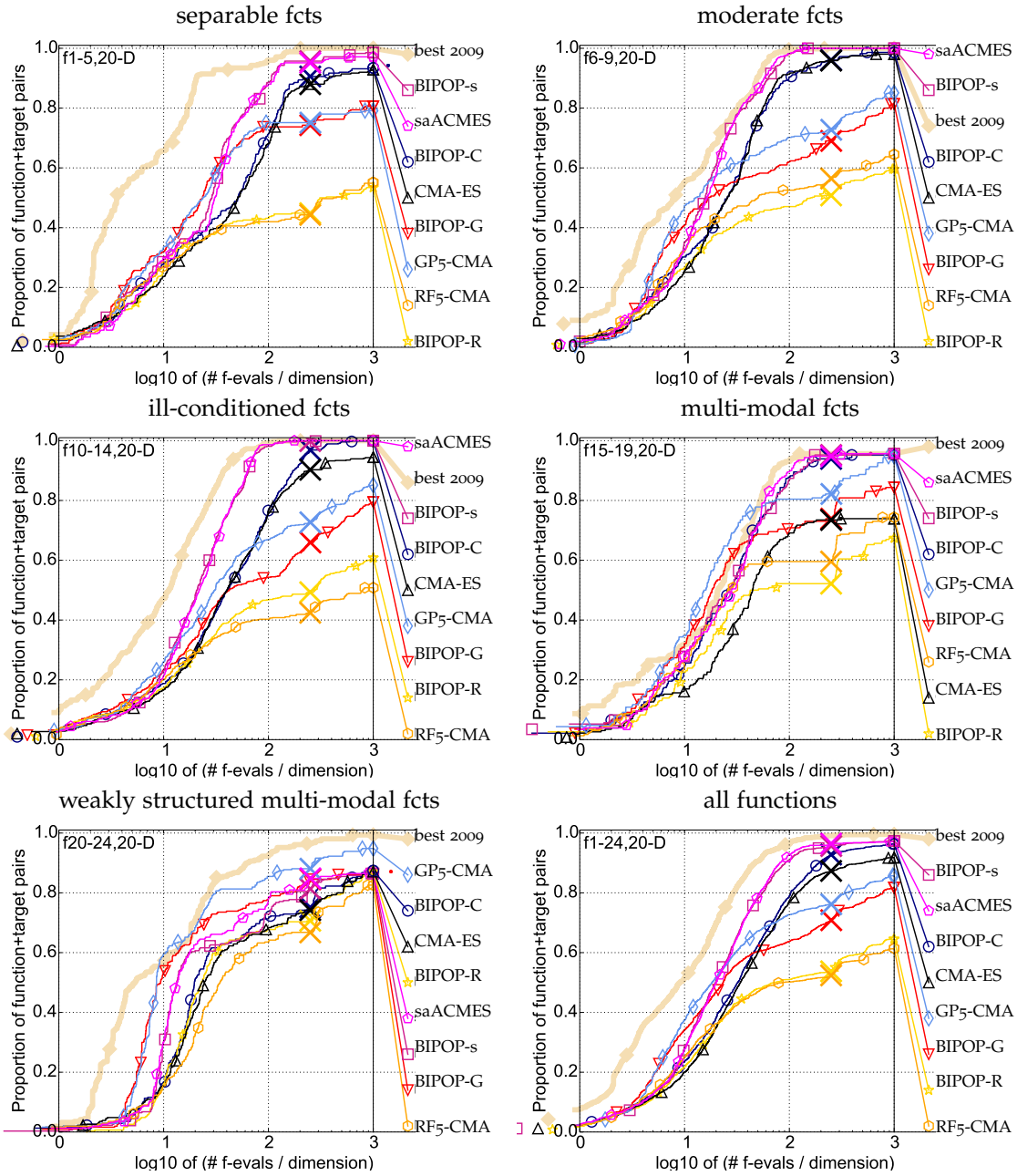


Figure 10: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for all functions and subgroups in $20D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times D$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" curve corresponds to the best ERT observed during BBOB 2009 for each selected target.

in the S-CMA-ES algorithm and comparing its speed-up using different settings against the CMA-ES without the surrogate model. Both the S-CMA-ES models mostly outperform the CMA-ES in initial phases of the optimization process. However, the speed-up is decreasing with the increasing number of evaluations. Gaussian processes outperformed random forests both in measuring the regression capabilities and using the models in the CMA-ES. On the other hand, the CMA-ES with random forests surrogate still exhibit speed-up on some benchmarks compared to the non-surrogate version.

The comparison of both surrogate-assisted versions with ^{s*}ACM-ES-k algorithm based on ordinal regression by Ranking SVR, and the original CMA-ES, all in their IPOP and BIPOP versions has shown that Gaussian process S-CMA-ES usually outperforms the ordinal-based ^{s*}ACM-ES-k in early stages of the algorithm search, especially on multimodal functions. The BIPOP versions of the algorithms did not increase the performance of appropriate IPOP versions except the Linear slope function f_5.

4.3 Adaptive surrogate-CMA-ES

In (Repický et al., 2017), we have presented an adaptive improvement for S-CMA-ES based on a general procedure introduced by Loshchilov et al. (2012) with the ^{s*}ACM-ES algorithm, in which the number of generations using the surrogate model before retraining is adjusted depending on the performance of the last instance of the surrogate. We have evaluated three algorithms differing in the measure of the surrogate model's performance on the COCO/BBOB framework.

4.3.1 *Adaptive Evolution Control for Surrogate CMA-ES*

The generation-based evolution strategy optimizes the fitness function and the surrogate model thereof in certain proportion. On problem areas that can be approximated well, a surrogate-assisted optimization might benefit from frequent utilization of the model, while on areas that are hard for the surrogate to approximate, frequent utilization of the model might degenerate the performance due to the model's inaccuracy.

Adaptation of the number of model evaluated generations $g_{\mathcal{M}}$ (in addition to other surrogate model parameters that we have not investigated in (Repický et al., 2017)) depending on the previous model's error has been proposed in ^{s*}ACM-ES by Loshchilov et al. (2012).

Let g be a generation that is marked as original-fitness-evaluated, and a newly-trained surrogate model $\mathcal{M}$. If $\mathcal{M}$ is the first surrogate trained so far, put $g_{\mathcal{M}} = 1$. Otherwise, an error ϵ of a previous surrogate model $\mathcal{M}^{\text{last}}$ is estimated on the newly evaluated population $(\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)})$ (see Algorithm 7). The error ϵ is then mapped into a number of consecutive generations $g_{\mathcal{M}}$, $g_{\mathcal{M}} \in [0, g_{\mathcal{M}}^{\max}]$, for which the surrogate $\mathcal{M}$ will be used (see Algorithm 8).

We investigated three approaches for expressing surrogate model error. As the CMA-ES depends primarily on the ranking of candidate solutions, the first two approaches, *Kendall correlation coefficient* and *Rank difference* are based on ranking. The third one, previously proposed by Loshchilov et al. (2013c), uses *Kullback-Leibler divergence* a. k. a. *information gain* to measure a difference between a multivariate normal distribution estimated from the fitness values $\mathbf{y}$ and a multivariate normal distribution estimated for the predicted values $\hat{\mathbf{y}}$.

Algorithm 7 Model error estimation in adaptive S-CMA-ES

Input: error type $\text{err} \in \{\text{err}_{\text{Kendall}}, \text{err}_{\text{RDE}}, \text{err}_{\text{KL}}\}$, CMA-ES generation number g, a newly sampled population $\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}$, fitness values and model predictions in generation g $\mathbf{y}, \hat{\mathbf{y}}$, CMA-ES variables $\mathbf{m}^{(g)}, \sigma^{(g)}, \mathbf{C}^{(g)}$, maximal error so far $\epsilon_{\max}$

1: **if** $\text{err} = \text{err}_{\text{Kendall}}$ **then**
2: $\quad \tau \leftarrow \text{Kendall}(\mathbf{y}, \hat{\mathbf{y}})$
3: $\quad \epsilon \leftarrow \frac{1}{2}(1 - \tau)$
4: **else if** $\text{err} = \text{err}_{\text{RDE}}$ **then**
5: $\quad \epsilon \leftarrow \text{err}_{\text{RDE}}(\mathbf{y}, \hat{\mathbf{y}})$
6: **else if** $\text{err} = \text{err}_{\text{KL}}$ **then**
7: $\quad (\mathbf{m}^{(g+1)}, \sigma^{(g+1)}, \mathbf{C}^{(g+1)}) \leftarrow$ CMA-ES update $((\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}), \mathbf{y}, \mathbf{m}^{(g+1)}, \sigma^{(g+1)}, \mathbf{C}^{(g+1)})$
8: $\quad (\mathbf{m}_{\mathcal{M}}^{(g+1)}, \sigma^{(g+1)}, \mathbf{C}_{\mathcal{M}}^{(g+1)}) \leftarrow$ CMA-ES update $((\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}), \hat{\mathbf{y}}, \mathbf{m}^{(g+1)}, \sigma^{(g+1)}, \mathbf{C}^{(g+1)})$
9: $\quad \epsilon \leftarrow \text{err}_{\text{KL}}(\mathcal{N}(\mathbf{m}_{\mathcal{M}}^{(g+1)}, {\sigma^{(g+1)}}^2 \mathbf{C}_{\mathcal{M}}^{(g+1)}) \| \mathcal{N}(\mathbf{m}^{(g+1)}, {\sigma^{(g+1)}}^2 \mathbf{C}^{(g+1)}))$
10: $\quad$ **if** $\epsilon > \epsilon_{\max}$ **then**
11: $\quad\quad \epsilon_{\max} \leftarrow \epsilon$
12: $\quad$ **end if**
13: $\quad \epsilon \leftarrow \frac{\epsilon}{\epsilon_{\max}}$ {*normalize in proportion to the historical maximum*}
14: **end if**

Output: estimated error value $\epsilon \in [0, 1]$

KENDALL RANK CORRELATION COEFFICIENT Kendall rank correlation coefficient τ measures similarity between two different orderings of the same set. Let $\mathbf{y} = (\text{bb}(\mathbf{x}_1), \ldots, \text{bb}(\mathbf{x}_\lambda))$ and $\hat{\mathbf{y}} = \left(\mathcal{M}^{\text{last}}(\mathbf{x}_1), \ldots, \mathcal{M}^{\text{last}}(\mathbf{x}_\lambda)\right)$ be the sequences of the fitness values and the predicted values of a population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$, respectively. A pair of indices (i, j), such that $i \neq j$, $i, j \in \{1, \ldots, \lambda\}$, is said to be concordant, if both $y_i < y_j$ and $\hat{y}_i < \hat{y}_j$ or if both $y_i > y_j$ and $\hat{y}_i > \hat{y}_j$. A discordant pair $(i, j), i \neq j$, $i, j \in \{1, \ldots, \lambda\}$ is one fulfilling that both $y_i < y_j$ and $\hat{y}_i > \hat{y}_j$ or both $y_i > y_j$ and $\hat{y}_i < \hat{y}_j$. Let n_c and d_c denote the number of concordant and discordant pairs of indices from $\{1, \ldots, \lambda\}$, respectively. The Kendall correlation coefficient τ between vectors $\mathbf{y}$ and $\hat{\mathbf{y}}$ is defined as:

$$\tau = \frac{2}{\lambda(\lambda - 1)}(n_c - n_d).$$

In the corresponding branch of Algorithm 7, the value τ is decreasingly scaled into interval $[0, 1]$.

RANKING DIFFERENCE ERROR In accordance with most of the model-assessment methods for the CMA-ES surrogates, we have also used the difference in ranking of the fitness values in a population as a measure of model quality. We have proposed a normalized version of the error measure used by Kern et al. (2006), which we call the *ranking difference error* (RDE). We denote as *ranking* a function $\rho : \mathbb{R}^\lambda \to \{1, \ldots, \lambda\}^\lambda$ which maps each element in a vector $\mathbf{y} \in \mathbb{R}^\lambda$ to its rank, i. e., $(\rho(\mathbf{y}))_i \leq (\rho(\mathbf{y}))_j$ whenever $(\mathbf{y})_i \leq (\mathbf{y})_j$. The RDE is defined for two vectors of fitness values $\mathbf{y}_1^*$, $\mathbf{y}_2^*$. We assume the second vector $\mathbf{y}_2^*$ to be a more accurate prediction: $\mathbf{y}_2^*$

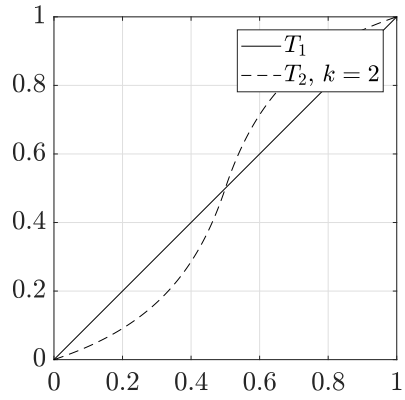


Figure 11: Model error transfer functions in adaptive S-CMA-ES

might contain more original-evaluated bb-values, or the $\mathbf{y}_2^*$'s model training set could be larger. The RDE is then calculated as

$$\mathrm{err}_{\mathrm{RDE}}(\mathbf{y}_1^*, \mathbf{y}_2^*) = \frac{\sum_{i:(\rho(\mathbf{y}_2^*))_i \leq \mu} |(\rho(\mathbf{y}_2^*))_i - (\rho(\mathbf{y}_1^*))_i|}{\max_{\pi \in \text{Permutations of } (1,\ldots,\lambda)} \sum_{i:\pi(i)\leq\mu} |\, i - \pi(i)|} \in [0,1]\,. \tag{67}$$

The larger the difference in rankings of these two bb-values vectors, the more the CMA-ES would be fooled if it got the less accurate bb-values prediction $\mathbf{y}_1^*$ instead of $\mathbf{y}_2^*$.

KULLBACK-LEIBLER DIVERGENCE Kullback-Leibler divergence from a continuous random variable Q with probability density function q to a continuous random variable P with probability density function p is defined as:

$$\mathrm{err}_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\, \ln \frac{p(x)}{q(x)}\, \mathrm{d}x\,. \tag{68}$$

For two multivariate normal distributions $\mathcal{N}_1(\mu_1, \Sigma_1)$ and $\mathcal{N}_2(\mu_2, \Sigma_2)$ with the same dimension D, the Kullback-Leibler divergence from $\mathcal{N}_2$ to $\mathcal{N}_1$ is:

$$\mathrm{err}_{\mathrm{KL}}(\mathcal{N}_1\|\mathcal{N}_2) = \frac{1}{2}\left(\mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + \ln\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1) - k\right). \tag{69}$$

The algorithm of model error estimation (see pseudocode in Algorithm 7) in generation g computes Kullback-Leibler divergence from a CMA-estimated multivariate normal distribution $\mathcal{N}(\mathbf{m}^{(g+1)}, {\sigma^{(g+1)}}^2\mathbf{C}^{(g+1)})$ w. r. t. fitness values $\mathbf{y}$ to a CMA-estimated multivariate normal distribution $\mathcal{N}(\mathbf{m}_{\mathcal{M}}^{(g+1)}, {\sigma_{\mathcal{M}}^{(g+1)}}^2\mathbf{C}_{\mathcal{M}}^{(g+1)})$ w. r. t. predicted values $\hat{\mathbf{y}}$. The result is normalized by the historical maximum (Step 13).

ADJUSTING THE NUMBER OF MODEL GENERATIONS The model of dependence of the number of consecutive model generations $g_{\mathcal{M}}$ on the model error (Algorithm 8) is almost identical to the approach used by Loshchilov et al. (2012). The history of surrogate model errors ϵ is exponentially smoothed with a rate r_u (Step 1). The error is truncated at a threshold ϵ_T so that resulting $g_{\mathcal{M}} = g_{\mathcal{M}}^{\max}$ for all values $\epsilon \geq \epsilon_T$ (Step 3). In contrast to Loshchilov et al. (2012), we

Algorithm 8 Updating the number of model generations in adaptive S-CMA-ES

Input: estimation of surrogate model error $\epsilon \in [0,1]$), a threshold at which the error is truncated to 1 $\epsilon_T \in [0,1]$, transfer function $\gamma\colon [0,1] \to [0,1]$, error update rate r_u, model error from the previous iteration ϵ_{last}, upper bound for admissible number of model generations $g_{\mathcal{M}}^{\max}$

1: $\epsilon \leftarrow (1 - r_u)\epsilon_{\text{last}} + r_u\epsilon$ {*exponential smoothing*}
2: $\epsilon_{\text{last}} \leftarrow \epsilon$
3: $\epsilon \leftarrow \frac{1}{\epsilon_T} \min\{\epsilon, \epsilon_T\}$ {*truncation to* 1}
4: $g_{\mathcal{M}} \leftarrow \text{round}(\gamma(1-\epsilon) g_{\mathcal{M}}^{\max})$ {*scaling into the admissible interval*}

Output: $g_{\mathcal{M}}$ — updated number of model-evaluated generations

consider two different transfer functions $T_1, T_2\colon [0,1] \to [0,1]$ (plotted in Figure 11) that scale the error into the admissible interval $[0, g_{\mathcal{M}}^{\max}]$:

$$T_1(x) = x \tag{70}$$

$$T_2(x;k) = \frac{\left(x - \frac{1}{2}\right)\left(1 + \frac{1}{k}\right)}{\left|2\left(x - \frac{1}{2}\right)\right| + \frac{1}{k}} + \frac{1}{2},\ k > 0. \tag{71}$$

Both functions are defined on $[0,1]$, moreover, $T_i(0) = 0$ and $T_i(1) = 1$ for $i = 1, 2$. Transfer function T_2 is a simple sigmoid function defined to be slightly less sensitive near the edges than in the middle. More control can thus be achieved in the regions of low and high error values. The parameter k determines the steepness of the sigmoid curve.

4.3.2 *Comparison of Adaptive and Static Versions of S-CMA-ES*

In this section, we show the comparison of the three S-CMA-ES versions adjusting $g_{\mathcal{M}}$ using three surrogate model error measures on the noiseless part of the COCO/BBOB framework presented in (Repický et al., 2017). We restrict our attention to S-CMA-ES with Gaussian processes, since they outperformed random forest-based surrogates in (Bajer et al., 2015).

Experimental Setup

We have evaluated the proposed adaptive generation-based evolution control for the S-CMA-ES with three different surrogate model error measures on the noiseless part of the COCO/BBOB framework (Hansen et al., 2009b, 2012) and compared with the S-CMA-ES and CMA-ES on instances $1-5$ and $41-50$ in dimensions 2, 3, 5, 10, and 20. Each trial was terminated when the $\mathfrak{bbo}_{\text{opt}}$ was reached within a small tolerance $\Delta\mathfrak{bbo}_t = 10^{-8}$ or when a budget of 250 FE/D was depleted. The algorithms' settings are summarized in the following paragraphs.

CMA-ES The CMA-ES results in BBOB format were downloaded from the BBOB 2010 workshop archive [1]. The CMA-ES used in those experiments was in version 3.40β and utilized an IPOP restart strategy (Auger and Hansen, 2005). The default parameter values employed in the CMA-ES were identical to Bajer et al. (2015) described in Section 4.2.1.

1 http://coco.gforge.inria.fr/data-archive/bbob/2010/

Table 4: Discretization of the aS-CMA-ES parameters.

Parameter	Discretization
γ	T_1 (70), T_2 (71)
ϵ_T	$0.5, 0.9$
$g_{\mathcal{M}}$	$5, 10, 20$
r_u	$0.2, 0.5, 0.8$

S-CMA-ES The S-CMA-ES was tested with two numbers of model-evaluated generations, $g_{\mathcal{M}} = 1$ (GP-1) and $g_{\mathcal{M}} = 5$ (GP-5). All other S-CMA-ES settings were left as described in Section 4.2.1. If not mentioned otherwise, the corresponding settings of adaptive versions of the S-CMA-ES are as just stated.

In order to find the most promising settings for each considered surrogate error measure, a full factorial experiment was conducted on one half of the noiseless testbed, namely on functions f_i for $i \in \{2, 3, 6, 8, 12, 13, 15, 17, 18, 21, 23, 24\}$. The discretization of continuous parameters $(\gamma, \epsilon_T, g_{\mathcal{M}}^{\max}, r_u)$ is reported in Table 4. All possible combinations of the parameters were ranked on the 12 selected functions according to the lowest achieved Δf^{med} (see *Results* in this Section) for different numbers of function evaluations $\text{FE}/D = 25, 50, 125, 250$. The best settings were chosen according to the highest sum of 1^{st} rank counts. Ties were resolved according to the lowest sum of ranks. All of the best settings included maximum model-evaluated generations $g_{\mathcal{M}}^{\max} = 5$. The remainder of the winning values is summarized in the following paragraphs.

KENDALL CORRELATION COEFFICIENT (ADA-KENDALL) Transfer function $\gamma = T_2$, error threshold $\epsilon_T = 0.5$ and update rate $r_u = 0.2$.

RANKING DIFFERENCE ERROR (ADA-RD) The same, except transfer function was $\gamma = T_1$.

KULLBACK-LEIBLER DIVERGENCE (ADA-KL) Transfer function $\gamma = T_2$, error threshold $\epsilon_T = 0.9$ and update rate $r_u = 0.5$.

CPU Timing

Table 5: The time in seconds per function evaluation for the Adaptive S-CMA-ES.

Algorithm	$2D$	$3D$	$5D$	$10D$	$20D$
ADA-KL	0.38	0.26	0.34	0.69	3.36
ADA-Kendall	0.47	0.45	0.61	1.29	6.27
ADA-RD	0.57	0.60	0.71	1.63	7.90

In order to assess computational costs other than the number of function evaluations, we calculate CPU timing per function evaluation for each algorithm and each dimensionality. Each experiment was divided into jobs by dimensionalities, functions and instances. All jobs were

Table 6: Mean ranks of the CMA-ES, the S-CMA-ES and all aS-CMA-ES versions over the BBOB and the Iman-Davenport variant of the Friedman test for the 10 considered combinations of dimensionalities and evaluation budgets. The lowest value is highlighted in bold. Statistically significant results at the significance level $\alpha = 0.05$ are marked by an asterisk.

Dim	$2D$		$3D$		$5D$		$10D$		$20D$	
$\#\text{FEs}/\#\text{FE}_\text{T}$	$1/3$	1	$1/3$	1	$1/3$	1	$1/3$	1	$1/3$	1
CMA-ES	4.04	4.25	4.25	3.96	4.38	3.58	4.67	3.83	4.58	4.42
GP-1	3.38	3.94	4.21	3.62	3.92	4.02	3.69	3.92	3.54	3.27
GP-5	3.54	3.12	**2.83**	4.08	3.81	4.35	4.25	4.42	4.23	4.52
ADA-KL	3.23	**2.85**	3.29	3.44	3.69	3.73	3.60	4.04	3.15	3.65
ADA-Ken	3.98	3.46	3.25	**2.90**	2.90	2.96	**2.27**	**2.40**	**2.33**	**2.23**
ADA-RD	**2.83**	3.38	3.17	3.00	**2.31**	**2.35**	2.52	2.40	3.17	2.92
F_F	1.48	1.89	2.52*	1.67	4.47*	4.13*	7.82*	6.50*	5.35*	6.62*

run in a single thread on the Czech national grid MetaCentrum. The average time per function evaluation for each algorithm and each tested dimensionality is summarized in Table 5.

Results

We test the difference in algorithms' convergence for significance on the whole noiseless testbed with the non-parametric Friedman test (Demšar, 2006). The algorithms are ranked on each BBOB function with respect to medians of log-scaled minimal distance Δf from the function optimum, denoted as Δf^{med}, at a fixed budget of function evaluations.

To account for different optimization scenarios, the test is conducted separately for all considered dimensionalities of the input space and two function evaluation budgets, a higher and a lower one. Let $\#\text{FE}_\text{T}$ be the smallest number of function evaluations at which at least one algorithm reached the target, i. e., satisfied $\Delta f^{\text{med}} \leq \Delta f_t$, or $\#\text{FE}_\text{T} = 250D$ if the target has not been reached. We set the higher budget for the tests to $\#\text{FE}_\text{T}$ and the lower budget to $\frac{\#\text{FE}_\text{T}}{3}$.

Mean ranks from the Friedman test are given in Table 6. The critical value for the Friedman test is 2.29.

The mean ranks differ significantly for all tested scenarios except for both tested numbers of function evaluations in $2D$ and the higher tested number of function evaluations in $3D$. Starting from $5D$ upwards, the lowest mean rank is achieved either by ADA-Kendall or ADA-RD at both tested #FEs.

In order to show pairwise differences, we perform a pairwise $N \times N$ comparison of the algorithms' average ranks by the post-hoc Friedman test with the Bergmann-Hommel correction of the family-wise error (García and Herrera, 2008) in cases when the null hypothesis of equal algorithms' performance was rejected. To better illustrate algorithms differences, we also count the number of benchmark functions at which one algorithm achieved a higher rank than the other. The pairwise score and the statistical significance of the pairwise mean rank differences are reported in Table 7. In the post-hoc test, ADA-Kendall significantly outperforms both the CMA-ES and GP-5 in $10D$ and $20D$. It also significantly outperforms GP-1 in $10D$ at the higher tested #FEs.

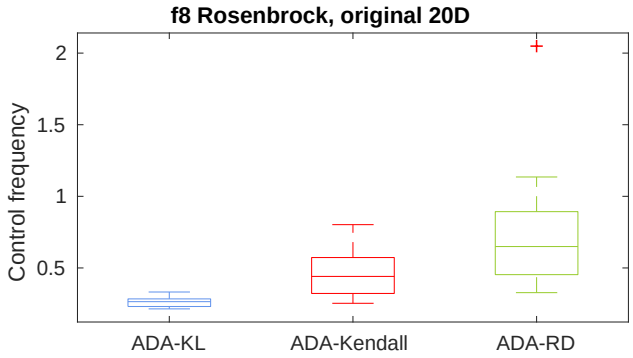


Figure 12: Average control frequency (the ratio of the number of total original-fitness-evaluated generations to the number of total model-evaluated generations) in aS-CMA-ES measured in 15 trials of each algorithm on f_8 in $20D$.

For illustration, the average control frequency given by the ratio of the number of total original-fitness-evaluated generations to the number of total model-evaluated generations within one trial, for data from 15 trials on Rosenbrock's function f_8 in $20D$ is given in Figure 12. The algorithm ADA-KL led to generally lower control frequencies than its competitors, which might explain its slightly inferior performance. Similar results were observed for the remaining functions and dimensionalities.

The cases when ADA-Kendall and ADA-RD were able to switch between more exploitation-oriented and more data-gathering-oriented behaviour can be studied on the results from COCO's postprocessing. GP-5 outperforms both GP-1 and the CMA-ES on the lower and middle parts of the empirical distribution functions basically for all dimensionalities (Figure 13). On the other hand, GP-1 outperforms GP-5 especially in later phases of the search (Figure 13).

The ability of ADA-Kendall and ADA-RD to switch to a less-exploitation mode when appropriate is eminent on the ECDFs plots in $20D$ (see Section 2.6.1), especially, on the moderate and the all-function groups (top right and bottom right on Figure 13), with exception of the well structured multimodal group (middle right), when they fail in the middle part and the weakly structured multimodal group (bottom left), when they fail towards the end of the search.

4.3.3 *Conclusion*

We have implemented several modifications of the S-CMA-ESconsidering three measures of surrogate model error according to which an adequate number of upcoming model-evaluated generations could be estimated online. We have compared three resulting algorithms on the COCO/BBOB framework with the S-CMA-ES parametrized by two different numbers of consecutive model-evaluated generations. The presented results summarize the performance of all compared algorithms on the whole noiseless part of the BBOB framework or its function groups. We have found two error measures, the Kendall rank correlation and the ranking difference error, that significantly outperformed the S-CMA-ES used with a higher number of model-evaluated generations, especially in higher dimensionalities of the input space. However, both of these algorithms provided only a minor improvement of the S-CMA-ES used with

Table 7: A pairwise comparison of the algorithms in $2D$, $3D$, $5D$, $10D$ and $20D$ over the BBOB for 2 different evaluation budgets. The comparison is based on medians over runs on 15 instances for each of all the 24 functions. 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 achieving a significantly lower value of the objective function than the column algorithm according to the Friedman post-hoc test with the Bergmann-Hommel correction at family-wise significance level $\alpha = 0.05$.

2D	CMA-ES		GP-1		GP-5		ADA-KL		ADA-Ken		ADA-RD	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
CMA-ES	—	—	8	8	11	8	10	7	11	10	7	9
GP-1	16	16	—	—	12	9	13	8	13	9	9	6
GP-5	13	16	12	15	—	—	11	10	14	13	9	14
ADA-KL	14	17	11	15	13	13	—	—	16	14	12	15
ADA-Ken	13	14	11	15	10	10	8	9	—	—	7	11
ADA-RD	17	15	15	17	15	9	12	9	17	12	—	—

3D	CMA-ES		GP-1		GP-5		ADA-KL		ADA-Ken		ADA-RD	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
CMA-ES	—	—	11	9	7	13	8	10	9	9	7	8
GP-1	13	15	—	—	7	14	7	11	6	8	10	9
GP-5	17	11	17	10	—	—	15	9	13	6	14	9
ADA-KL	16	14	17	13	9	15	—	—	13	11	10	9
ADA-Ken	15	15	18	16	11	17	11	13	—	—	11	12
ADA-RD	17	16	14	15	10	14	14	15	13	10	—	—

5D	CMA-ES		GP-1		GP-5		ADA-KL		ADA-Ken		ADA-RD	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
CMA-ES	—	—	8	12	11	14	11	14	7	10	2	8
GP-1	16	12	—	—	11	12	11	12	9	7	3	4
GP-5	13	10	13	12	—	—	10	6	9	5	8	7
ADA-KL	13	10	13	11	14	18	—	—	7	10	8	5
ADA-Ken	17	14	15	17	15	19	17	14	—	—	10	9
ADA-RD	22*	16	21*	20*	16*	17*	16	19	14	14	—	—

10D	CMA-ES		GP-1		GP-5		ADA-KL		ADA-Ken		ADA-RD	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
CMA-ES	—	—	7	12	13	14	10	14	1	8	1	4
GP-1	17	12	—	—	15	15	14	14	4	5	5	4
GP-5	11	10	9	9	—	—	8	11	6	4	8	5
ADA-KL	14	10	10	10	16	13	—	—	8	5	9	8
ADA-Ken	23*	16*	20	19*	18*	20*	16	19*	—	—	13	12
ADA-RD	23*	20*	19	20*	16*	19*	15	15*	11	12	—	—

20D	CMA-ES		GP-1		GP-5		ADA-KL		ADA-Ken		ADA-RD	
$^{\#FEs}/_{\#FE_T}$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
CMA-ES	—	—	7	5	11	12	9	13	4	3	3	5
GP-1	17	19	—	—	14	17	12	16	7	6	9	8
GP-5	13	12	10	7	—	—	4	5	6	5	10	6
ADA-KL	15	11	12	8	20	19	—	—	9	8	13	10
ADA-Ken	20*	21*	17	18	18*	19*	15	16	—	—	17	17
ADA-RD	21	19*	15	16	14	18*	11	14	7	7	—	—

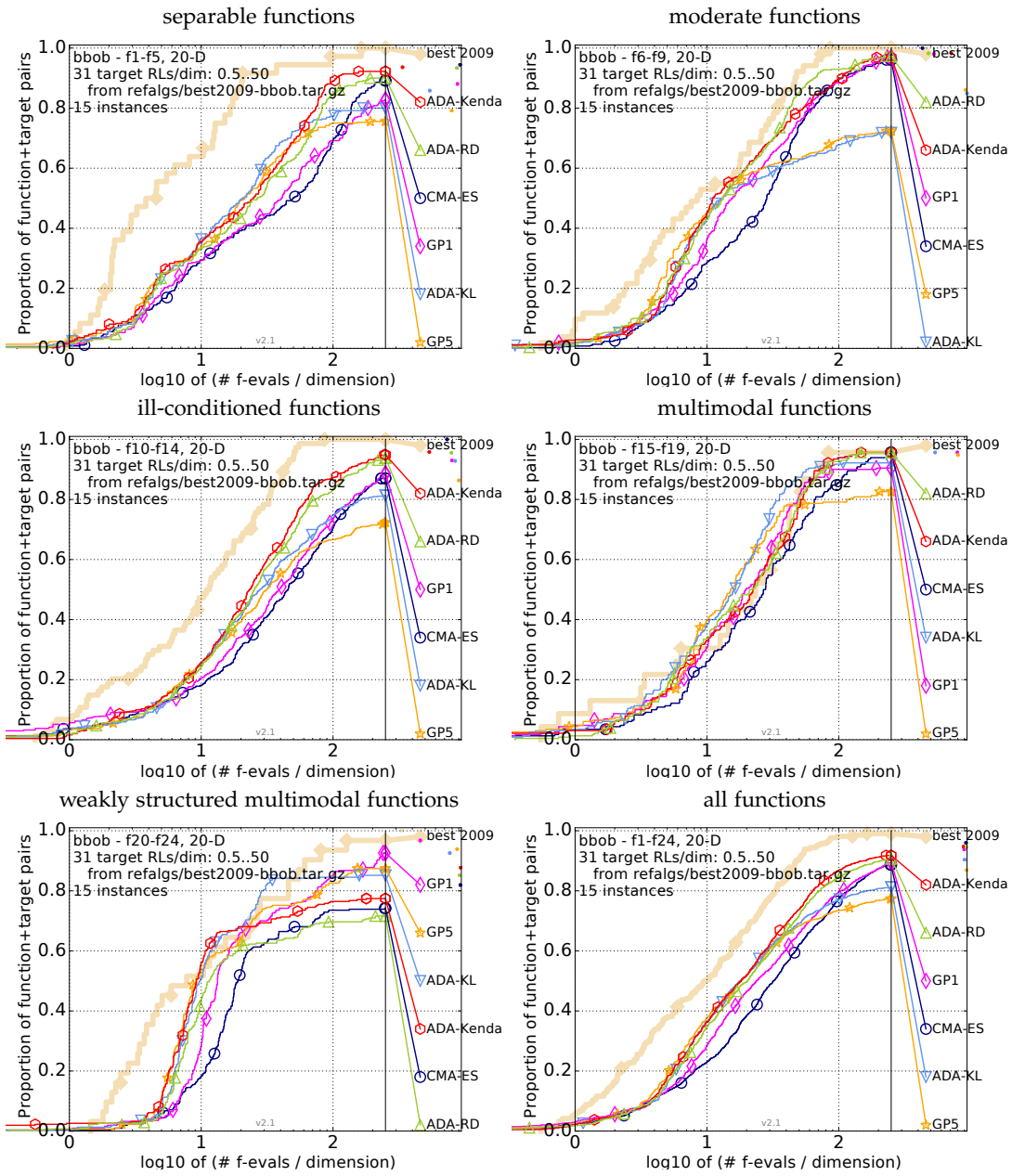


Figure 13: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension FE/D for all functions and subgroups in 20D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times D$, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 50\}$. The "best 2009" curve corresponds to the best average running time observed during BBOB 2009 for each selected target. Legend: ○: CMA-ES, ◇: GP1, ☆: GP5, ▽: ADA-KL, ⬡: ADA-Kendall, △: ADA-RD

a lower number of model-evaluated generations and in some tested scenarios fell behind both tested settings of the S-CMA-ES.

4.4 DOUBLY TRAINED S-CMA-ES

The DTS-CMA-ES (Pitra et al., 2016) is the S-CMA-ES successor replacing the generation evolution control by the *doubly trained evolution control*, which utilizes the ability of Gaussian processes to provide the distribution of predicted points.

DOUBLY TRAINED EVOLUTION CONTROL In the generation-based EC, the whole population is always original-evaluated at once which makes the exploitation of the surrogate model predictive uncertainty difficult. Additionally, the CMA-ES can drift outside the region where any archive points are stored. These two facts led us to employ another evolution control. However, the other common choice, individual-based EC, can modify the distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$ of the CMA-ES population due to non-uniform selection of the λ promising points from the larger set of model-evaluated points.

Although Loshchilov et al. (2010) suggested subsampling the points according to the predicted fitness mapped onto a Gaussian distribution and Hansen (2011) has later given restrictions on injecting external points into the CMA-ES population, we have proposed another solution called *doubly trained* EC in (Pitra et al., 2016). Each generation of this EC can be summarized in the following steps:

(1) sample a new population of size λ (standard CMA-ES offspring),
(2) train the *first* surrogate model on the original-evaluated points from the archive $\mathcal{A}$,
(3) select $\lceil \alpha\lambda \rceil$ point(s) wrt. a criterion $\mathfrak{C}$, which is based on the *first* model's prediction,
(4) evaluate these point(s) with the original fitness,
(5) re-train the surrogate model also using these new point(s), and
(6) predict the fitness for the non-original evaluated points with this *second* model.

The algorithm is partially similar to the evolution control of the lmm-CMA, but the lmm-CMA always selects the points with the best-predicted fitness (in step 3) and retrains the model generally more times than twice (it cycles steps 3–5).

The key characteristics of the doubly trained EC are the following. First, due to step 1, the population is always a sample from the CMA-ES distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$, similarly to the generation-based EC. Second, in step 3, the Gaussian process estimation of uncertainty can be utilized for the selection of a typically low number of original-evaluated points, see below for possible criteria and the description of the parameter α. Third, the estimated fitness values in step 6, which are returned to the CMA-ES, are predicted using the model trained on as recent points as possible. Finally, evaluating at least a small number of points in each generation using the original fitness maintains at least a decent number of points in the training set near the current CMA-ES distribution mean.

Employing the doubly trained EC in the CMA-ES results in the DTS-CMA-ES (Pitra et al., 2016) whose pseudocode is shown in Algorithm 9. The algorithm is parametrized by the parameter $\alpha^{(0)}$ — the initial value of the ratio of original-evaluated points in a population, by the criterion $\mathfrak{C}$ for the selection of these points, and by the surrogate model and its parameters. Furthermore, a user can let the ratio α adapt throughout the algorithm run, which requires an additional set of parameters — see Algorithm 10 and the corresponding paragraph below.

The DTS-CMA-ES is analysed in more detail from the EC point of view in Section 4.8.1.

Algorithm 9 DTS-CMA-ES

Input: original fitness function $\mathfrak{bb}$, step-size $\sigma^{(0)} \in \mathbb{R}_+$, initial mean $\mathbf{m}^{(0)} \in \mathbb{R}^D$, initial ratio of original-evaluated points $\alpha^{(0)}$, criterion for the selection of original-evaluated points $\mathfrak{C}$, self-adaptation parameters β, $\epsilon^{(0)}$, $\epsilon_{\min}$, $\epsilon_{\max}$, $\alpha_{\min}$, $\alpha_{\max}$ (see Alg. 10), training set selection method TSS, surrogate model parameters ψ (see Alg. 5)

$\mathcal{A} \leftarrow \emptyset$; $\sigma^{(0)}, \mathbf{m}^{(0)}, \mathbf{C}^{(0)} \leftarrow$ CMA-ES initialize {*initialization*}
for generation $g = 0, 1, 2, \ldots$ until stopping conditions met **do**
  $\mathbf{x}_k \sim \mathcal{N}(\mathbf{m}^{(g)}, {\sigma^{(g)}}^2 \mathbf{C}^{(g)})$ for $k = 1, \ldots, \lambda$ {*CMA-ES sampling*}
  $\mathcal{M}_1 \leftarrow \text{trainModel}(\mathcal{A}, \text{TSS}, N^{\mathcal{M}}_{\min}, \psi, \mathbf{m}^{(g)}, \sigma^{(g)}, \mathbf{C}^{(g)})$ {1^{st} *model training*}
  $(\hat{y}, s^2) \leftarrow \mathcal{M}_1([\mathbf{x}_1, \ldots, \mathbf{x}_\lambda])$ {*model-fitness evaluation*}
  $\mathbf{X}_{\text{orig}} \leftarrow$ select $\lceil \alpha^{(g)} \lambda \rceil$ best points according to the criterion $\mathfrak{C}$
  $\mathbf{y}_{\text{orig}} \leftarrow \mathfrak{bb}(\mathbf{X}_{\text{orig}})$ {*original-fitness evaluation*}
  $\mathcal{A} = \mathcal{A} \cup \{(\mathbf{X}_{\text{orig}}, \mathbf{y}_{\text{orig}})\}$ {*archive update*}
  $\mathcal{M}_2 \leftarrow \text{trainModel}(\mathcal{A}, \text{TSS}, N^{\mathcal{M}}_{\min}, \psi, \mathbf{m}^{(g)}, \sigma^{(g)}, \mathbf{C}^{(g)})$ {2^{nd} *model training*}
  $\mathbf{y} \leftarrow \mathcal{M}_2([\mathbf{x}_1, \ldots, \mathbf{x}_\lambda])$ {2^{nd} *model prediction*}
  $(\mathbf{y})_k \leftarrow (\mathbf{y}_{\text{orig}})_i$ for all original-evaluated $(\mathbf{y}_{\text{orig}})_i \in \mathbf{y}_{\text{orig}}$ {*fitness replace*}
  $(\alpha^{(g+1)}, \epsilon^{(g+1)}) \leftarrow \text{selfAdaptation}(\epsilon^{(g)}, \hat{\mathbf{y}}, \mathbf{y}; \beta, \epsilon_{\min}, \epsilon_{\max}, \alpha_{\min}, \alpha_{\max})$ {*Alg. 10*}
  sorted $\mathbf{x}_{1:\lambda} \leftarrow$ sort $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ based on $(\mathbf{y}_1, \ldots, \mathbf{y}_\lambda)^\top$ {*population sort*}
  $\sigma^{(g+1)}, \mathbf{m}^{(g+1)}, \mathbf{C}^{(g+1)} \leftarrow$ CMA-ES update based on $\mathbf{x}_{1:\lambda}$
end for
$\hat{\mathbf{x}}^{\text{opt}} \leftarrow \mathbf{x}_k$ from $\mathcal{A}$ corresponding to the minimal y_k

Output: $\hat{\mathbf{x}}^{\text{opt}}$ – point with the minimum achieved fitness

CRITERIA FOR THE SELECTION OF ORIGINAL-EVALUATED POINTS. While the surrogate algorithms assisted by models not based on Gaussian processes select points for the original evaluation mostly according to the predicted fitness, the Gaussian process surrogates, which for any point $\mathbf{x}$ predict the whole Gaussian distribution $\mathcal{N}(\hat{y}(\mathbf{x}), (\hat{s}(\mathbf{x}))^2)$, offer more options when used with the individual-based or doubly trained evolution control. The following criteria are considered in the DTS-CMA-ES. Whereas the first four are defined for any point of the input space and have been used in Bayesian optimization for decades, the last one, *expected ranking difference error* (ERDE), is our new contribution directly exploiting the DTS-CMA-ES' Gaussian processes prediction and is defined only for the points from the considered population.

- *GP predictive mean*. The GP mean prediction $\hat{y}(\mathbf{x})$ is the maximum-likelihood estimate of the original fitness. This criterion is defined as its negative value

$$\mathfrak{C}_{\text{M}}(\mathbf{x}) = -\hat{y}(\mathbf{x}) . \tag{72}$$

- *GP predictive standard deviation*. Choosing the points with the highest uncertainty leads to the criterion

$$\mathfrak{C}_{\text{STD}}(\mathbf{x}) = \hat{s}(\mathbf{x}) .$$

- *Expected Improvement* (EI). If $y_{\min}$ stands for the minimum fitness in the considered training set $y_1, \ldots, y_N$, the EI criterion is

$$\mathfrak{C}_{\text{EI}}(\mathbf{x}) = E((y_{\min} - \hat{\mathfrak{bb}}(\mathbf{x}))\mathbb{I}(\hat{\mathfrak{bb}}(\mathbf{x}) < y_{\min}) \,|\, y_1, \ldots, y_N) , \text{ where} \tag{73}$$

$$\mathbb{I}(\hat{\mathfrak{bb}}(\mathbf{x}) < y_{\min}) = \begin{cases} 1 & \text{for } \hat{\mathfrak{bb}}(\mathbf{x}) < y_{\min} \\ 0 & \text{for } \hat{\mathfrak{bb}}(\mathbf{x}) \geq y_{\min} \end{cases} .$$

◇ *Probability of Improvement* (PoI). The PoI express the probability of finding lower fitness than some threshold T

$$\mathfrak{C}_{\text{PoI}}(\mathbf{x}, T) = P(\hat{\mathfrak{bb}}(\mathbf{x}) \leq T \,|\, y_1, \ldots, y_N) = \Phi\left(\frac{T - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) \tag{74}$$

where Φ is the distribution function of $\mathcal{N}(0,1)$. As T, the value $T = y_{\min}$ or a slightly higher value is usually chosen, see, e. g., (Jones, 2001) for an evaluation of these thresholds. This criterion was shown to provide higher speed up than EI when used with the DTS-CMA-ES.

◇ *Expected ranking difference error* ERDE. The goal of this criterion is to select the points from the current population for which the expected RDE would decrease most after adding them to the GP training set. It is a ranking counterpart of the GP predictive standard deviation criterion since it selects the points for which the model is least certain. In DTS-CMA-ES, the err_{RDE} has been used as an error measure[2].

The trained Gaussian process $\mathcal{M}_1$ estimates the fitness distribution for each point in the population as $\mathcal{N}(\hat{y}_k, (\hat{s}_k)^2)$, where $\hat{y}_k$ and $\hat{s}_k$ are given from equations (19) and (20).

For each point $\mathbf{x}_k^* \in \{\mathbf{x}_1, \ldots, \mathbf{x}_\lambda\}$ in the considered population, let us denote $\mathbf{X}_{-k}$ the population with the k-th point removed and $\hat{\mathbf{y}}_{-k} = \mathcal{M}_1(\mathbf{X}_{-k})$ the vector of the corresponding mean predictions of the first DTS-CMA-ES model. The ERDE criterion is the expected contribution of $\mathbf{x}_k^*$ to the ranking error

$$\mathfrak{C}_{\text{ERDE}}(\mathbf{x}_k^*, \mathbf{X}_{-k}) = \mathbb{E}[\text{err}_{\text{RDE}}(\hat{\mathbf{y}}_{-k}, \hat{\mathbf{y}}_{-k}^+)] = \int_{-\infty}^{\infty} \text{err}_{\text{RDE}}(\hat{\mathbf{y}}_{-k}, \hat{\mathbf{y}}_{-k}^+)\, \varphi(y_k)\, \mathrm{d}y_k \tag{75}$$

where $\hat{\mathbf{y}}_{-k}^+ = \mathcal{M}_1^+(\mathbf{X}_{-k})$ is the vector of mean predictions of a GP $\mathcal{M}_1^+$. This process $\mathcal{M}_1^+$ is identical to the first model $\mathcal{M}_1$ except that the k-th point $(\mathbf{x}_k^*, y_k)$ is additionally incorporated into its training set, say to its end, which becomes $(\mathbf{X}_{N+}, \mathbf{y}_{N+}) = ([\mathbf{X}_N\, \mathbf{x}_k^*], (\mathbf{y}_N^\top\ y_k)^\top)$; particularly, $\mathcal{M}_1^+$ uses the same hyperparameters as $\mathcal{M}_1$. The expectation is calculated over normally distributed $y_k \sim \mathcal{N}(\hat{y}_k, (\hat{s}_k)^2)$ with density $\varphi(y_k)$, where $\hat{y}_k$ and $(\hat{s}_k)^2$ are defined using the first DTS-CMA-ES model $\mathcal{M}_1$ by equations (19) and (20).

The vector of predicted means $\hat{\mathbf{y}}_{-k}^+$ is given by (23) as

$$\hat{\mathbf{y}}_{-k}^+ \quad = \quad \kappa(\mathbf{X}_{-k}, \mathbf{X}_{N+})\, \mathbf{C}_{N+}^{-1}\, \mathbf{y}_{N+} \,.$$

For the matrix $\mathbf{A} = \kappa(\mathbf{X}_{-k}, \mathbf{X}_{N+})\mathbf{C}_{N+}^{-1}$ that is for a considered $\mathbf{x}_k^*$ constant, the value of each element of $\hat{\mathbf{y}}_{-k}^+$ depends only on y_k as

$$(\hat{\mathbf{y}}_{-k}^+)_i \quad = \quad (\mathbf{A})_{i,1\ldots N}\, \mathbf{y}_N + y_k\, (\mathbf{A})_{i,N+1} \,.$$

2 err_{RDE} could be replaced by arbitrary error measure between $\hat{\mathfrak{bb}}$-values vectors which only depends on ranks of the elements of these vectors.

Algorithm 10 selfAdaptation($\epsilon^{(g)}, \hat{\mathbf{y}}, \mathbf{y}; \beta, \epsilon_{\min}, \epsilon_{\max}, \alpha_{\min}, \alpha_{\max}$)

Input: smoothed error from the last generation $\epsilon^{(g)}$,
vector of the first model prediction $\hat{\mathbf{y}}$,
vector of the second model prediction with original evaluations $\mathbf{y}$,
update rate β, minimum and maximum ranking error: $\epsilon_{\min}$, $\epsilon_{\max}$,
min. and max. ratio of original-evaluated points: $\alpha_{\min}$, $\alpha_{\max}$

1: $\epsilon_{\text{RDE}} \leftarrow \text{err}_{\text{RDE}}(\hat{\mathbf{y}}, \mathbf{y})$ {*estimation of the model's error*}
2: $\epsilon^{(g+1)} \leftarrow (1-\beta)\epsilon^{(g)} + \beta\epsilon_{\text{RDE}}$ {*exponential smoothing of the error*}
3: $\alpha^{(g+1)} \leftarrow \alpha_{\min} + \max\{0, \min\{1, \frac{\epsilon^{(g+1)} - \epsilon_{\min}}{\epsilon_{\max} - \epsilon_{\min}}\}\} \cdot (\alpha_{\max} - \alpha_{\min})$ {*update of* α}

Output: $\alpha^{(g+1)}$ – ratio of original-evaluted points for the next generation,
$\epsilon^{(g+1)}$ – new smoothed error

Further, the integral (75) over y_k can be calculated as the sum of sub-integrals J_p over the intervals $I_p = [l_p, u_p]$, $p = 1, \ldots, (\frac{1}{2}(\lambda-1)(\lambda-2)+1)$ defined such that the ranking of $\hat{\mathbf{y}}^+_{-k}$ is constant. The intervals I_p's can be derived from the mutual comparisons

$$(\hat{\mathbf{y}}^+_{-k})_i \leq (\hat{\mathbf{y}}^+_{-k})_j \quad \text{for } i \neq j,\ i,j = 1, \ldots, (\lambda-1) \tag{76}$$

where the interval boundaries $u_1 = l_2 \leq u_2 = l_3 \leq \ldots \leq u_{p-1} = l_p$ are the points for which the equalities (76) arise. As the rankings are on each I_p constant, the sub-integrals simplify to

$$J_p = \text{err}_{\text{RDE}}(\hat{\mathbf{y}}_{-k}, \hat{\mathbf{y}}^+_{-k})\,(\Phi(u_p) - \Phi(l_p))$$

where Φ is the distribution function of $\mathcal{N}(\hat{y}_k, (\hat{s}_k)^2)$.

As the matrix $\mathbf{C}^{-1}_{N+1}$ and hence also $\mathbf{A}$ can be calculated in $\mathcal{O}(N^3)$ steps, the complexity of calculating ERDE for each point is $\mathcal{O}(N^3 + \lambda^2 N)$ which enables efficient computation of this criterion even for moderately large λ.

4.4.1 *Self-adaptation of the original-evaluated points ratio α*

Whereas the speed-up of all surrogate-assisted algorithms is based on the model's ability to estimate fitness, the models can mislead the optimization algorithm when their error is high. Similarly to the lmm-CMA and [s*]ACM-ES, such an adverse effect can be controlled by raising the number of original-evaluated points if necessary. In (Pitra et al., 2017a), we have investigated the usage of the model according to the model's error: The DTS-CMA-ES measures the ranking difference error of the model and raises the original-evaluated points ratio α if the model error gets higher. The details are summarized in Algorithm 10: first, the err_{RDE} of the last model is exponentially smoothed with the update rate β, and the ratio α is then calculated as the result of a linear transfer function that is defined by the bounds on the smoothed error $\epsilon_{\min}$, $\epsilon_{\max}$ and on the resulting ratio $\alpha_{\min}$, $\alpha_{\max}$.

Having analyzed the err_{RDE} error measures on the COCO testbed, we observed that the measured RDE error $\epsilon^{(g)}$ depends on the ratio α and the dimension D:

$$\epsilon_{\min} = f^{\epsilon_{\min}}(\alpha, D), \quad \epsilon_{\max} = f^{\epsilon_{\max}}(\alpha, D). \tag{77}$$

Especially dependence on α is not surprising: from the definition of err_{RDE} (67) follows that the more reevaluated points, the higher number of summands in nominator of (67) and hence

the higher err_{RDE} value. Due to mutual dependence of the parameters ϵ and α, the calculation of α in each generation is performed iteratively until convergence of α:

(1) calculate error thresholds ϵ_{min}, ϵ_{max} using the last used ratio α – either from the previous iteration, or from the previous generation (see Equation (77))

(2) calculate new ratio α using newly calculated ϵ_{min}, ϵ_{max} (see Step 3)

In our implementation, the functions $f^{\epsilon_{\text{min}}}$ and $f^{\epsilon_{\text{min}}}$ are results of multiple linear regression – see paragraph Linear Models for the details.

Experimental Setup for Evaluation of DTS Self-adaptivity

In (Pitra et al., 2017a), we compared the performances of the DTS-CMA-ES using selfAdaptation to adjust α (aDTS) to the original DTS-CMA-ES (Pitra et al., 2016) using fixed α, the CMA-ES (Hansen, 2006), and two other surrogate-assisted versions of the CMA-ES, the lmm-CMA (Auger et al., 2013; Kern et al., 2006) and the ^{s*}ACM-ES (Loshchilov et al., 2013a), on the noiseless part of the COCO framework (Hansen et al., 2009b, 2012).

The considered algorithms were evaluated using the 24 noiseless COCO single-objective benchmarks Hansen et al. (2012, 2009b) in dimensions $D = 2, 3, 5$ and 10 on 15 different instances per function. The functions were divided into three groups according to the difficulty of their modeling with a GP model, where two groups were used for tuning aDTS parameters and the remaining group was utilized to test the results of that tuning. The method of dividing the functions into those groups will be described below in connection with the aDTS settings. The experiment stopping criteria were reaching either the maximum budget of 250 function evaluations per dimension (FE/D), or reaching the target distance from the function optimum $\Delta f_T = 10^{-8}$. The following paragraphs summarize the parameters of the compared algorithms.

CMA-ES The original CMA-ES was tested in its IPOP-CMA-ES version (Matlab code v. 3.61) using settings identical to (Bajer et al., 2015) described in Section 4.2.1.

LMM-CMA The lmm-CMA was employed in its improved version published by Auger et al. (2013). The results have been downloaded from the COCO/BBOB results data archive [3] in its GECCO 2013 settings.

^{s*}ACM-ES We have used the bi-population version of the ^{s*}ACM-ES, the BIPOP-^{s*}ACM-ES-k by Loshchilov et al. (2013a). Similarly to the lmm-CMA, the algorithm results have also been downloaded from the COCO/BBOB results data archive[4].

STATIC DTS-CMA-ES The original DTS-CMA-ES was tested using the overall best settings from (Pitra et al., 2016): the prediction variance $\mathfrak{C}_{\text{STD}}$ of Gaussian process model as the uncertainty criterion, the population size $\lambda = 8 + \lfloor 6 \ln D \rfloor$, and the ratio of points evaluated by the original fitness $\alpha = 0.05$. The results of the DTS-CMA-ES are slightly different from results in (Pitra et al., 2016, 2017c) due to a correction of a bug in the original version which was affecting the selection of points to be evaluated by the original fitness using an uncertainty criterion.

3 http://coco.gforge.inria.fr/data-archive/2013/lmm-CMA-ES_auger_noiseless.tgz
4 http://coco.gforge.inria.fr/data-archive/2013/BIPOP-saACM-k_loshchilov_noiseless.tgz

ADAPTIVE DTS-CMA-ES The aDTS was tested with multiple settings of parameters. First, the linear regression models of lower and upper bounds for the error measure $\epsilon_{\min}$, $\epsilon_{\max}$ were identified via measuring $\mathrm{err}_{\mathrm{RDE}}$ on datasets from DTS-CMA-ES runs on the COCO/BBOB benchmarks.

As a first step, we figured out six BBOB functions which are the *easiest* and six which are the *hardest* to regress by our Gaussian process model based on the $\mathrm{err}_{\mathrm{RDE}}$ measured on 1250 *independent testsets* per function in each dimension: 10 sets of λ points in each of 25 equidistantly selected generations from the DTS-CMA-ES runs on the first 5 instances, see Table 8 for these sets of functions and their respective errors. The functions which were not identified as *easiest* or *hardest* form the *test* function set.

LINEAR MODELS Using the same 25 DTS-CMA-ES "snapshots" on each of 5 instances, we calculated medians (Q_2) and the third quartiles (Q_3) of *measured* $\mathrm{err}_{\mathrm{RDE}}$ on populations from both groups of *easiest* and *hardest* functions, where we used five different proportions of original-evaluated points $\alpha = \{0.04, 0.25, 0.5, 0.75, 1.00\}$ which were available for retrained models and thus also for measuring models' errors $\epsilon^{(g)}$. These quartiles were regressed by multiple linear regression models using stepwise regression from a full quadratic model of the ratio α and dimension D or its logarithm $\ln(D)$ (decision whether to use $\ln(D)$ or D was according to the root mean-squared error (RMSE) of the final stepwise models); the stepwise regression was removing terms with the highest p-value > 0.05. The coefficients $\epsilon^{Q2}_{\min}$ and $\epsilon^{Q3}_{\min}$ of the lower thresholds were estimated on the data from *easiest* functions and the coefficients $\epsilon^{Q2}_{\max}$ and $\epsilon^{Q3}_{\max}$ of the higher thresholds on the data from *hardest* functions, which resulted in the following models:

$$\epsilon^{Q2}_{\min}(\alpha,D) = (1 \quad \ln(D) \quad \alpha \quad \alpha\ln(D) \quad \alpha^2) \cdot \mathbf{b}_1\,, \tag{78}$$
$$\epsilon^{Q3}_{\min}(\alpha,D) = (1 \quad D \quad \alpha \quad \alpha D \quad \alpha^2) \cdot \mathbf{b}_2\,, \tag{79}$$
$$\epsilon^{Q2}_{\max}(\alpha,D) = (1 \quad D \quad \alpha \quad \alpha D \quad \alpha^2) \cdot \mathbf{b}_3\,, \tag{80}$$
$$\epsilon^{Q3}_{\max}(\alpha,D) = (1 \quad \ln(D) \quad \alpha \quad \alpha\ln(D) \quad \alpha^2) \cdot \mathbf{b}_4\,, \tag{81}$$

where

$$\mathbf{b}_1 = (0.11 \quad -0.0092 \quad -0.13 \quad 0.044 \quad 0.14)^\top\,, \tag{82}$$
$$\mathbf{b}_2 = (0.17 \quad -0.00067 \quad -0.095 \quad 0.0087 \quad 0.15)^\top\,, \tag{83}$$
$$\mathbf{b}_3 = (0.18 \quad -0.0027 \quad 0.44 \quad 0.0032 \quad -0.14)^\top\,, \tag{84}$$
$$\mathbf{b}_4 = (0.35 \quad -0.047 \quad 0.44 \quad 0.044 \quad -0.19)^\top\,. \tag{85}$$

For the remaining investigations, three different values of exponential smoothing update rate were used for comparison $\beta = \{0.3, 0.4, 0.5\}$. The minimal and maximal values of α were set to $\alpha_{\min} = 0.04$ and $\alpha_{\max} = 1.0$ because lower α values than 0.04 would yield to less than one original-evaluated point per generation, and the aDTS has to be able to spend the whole populations for the original evaluations in order to work well on functions where GP model is poor (e. g., on ATTRACTIVE SECTOR function f_6). The initial error and original ratio values were set to $\epsilon^{(0)} = 0.05$ and $\alpha^{(0)} = 0.05$. The rest of aDTS parameters were left the same as in the original DTS-CMA-ES settings.

Results of DTS-CMA-ES adaptivity validation

The results in Figures 15, 17, and 18 and in Table 10 show the effect of adaptivity implemented in the DTS-CMA-ES (see Section 2.6.2 for explanation of convergence graphs).

We have tested the statistical significance of differences in algorithms' performance on 12 COCO/BBOB *test* functions in $10D$ for separately two evaluation budgets using the Iman and Davenport's improvement of the Friedman test (Demšar, 2006). Let $\#\mathrm{FE}_\mathrm{T}$ be the smallest number of function evaluations on which at least one algorithm reached the target, i. e., satisfied $\Delta_f^{\mathrm{med}} \leq \Delta f_T$, or $\#\mathrm{FE}_\mathrm{T} = 250D$ if no algorithm reached the target within $250D$ evaluations. The algorithms are ranked on each COCO/BBOB *test* 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 $\#\mathrm{FEs} = \#\mathrm{FE}_\mathrm{T}$ ($p < 10^{-3}$), as well as at a lower budget $\#\mathrm{FEs} = \frac{\#\mathrm{FE}_\mathrm{T}}{3}$ ($p < 10^{-3}$).

We test pairwise differences in performance utilizing the post-hoc Friedman test (García and Herrera, 2008) with the Bergmann-Hommel correction controlling the family-wise error in cases when the null hypothesis of equal algorithms' performance was rejected. To illustrate algorithms' differences, the numbers of *test* functions at which one algorithm achieved a higher rank than the other are reported in Table 10. The table also contains the pairwise statistical significances.

We have compared the performances of aDTS using twelve settings differing in ϵ_{min}, ϵ_{max}, and β. Table 9 illustrates the counts of the 1st ranks of the compared settings according to the lowest achieved Δ_f^{med} for 25, 50, 100, and 200 FE/D respectively. The counts are summed across the testing sets of benchmark functions in each individual dimension.

Although the algorithm is rather robust to exact setting of smoothing update rate, we have found that the lower the β, the better the performance is usually observed (see Table 9), and thus the following experiments use the rate $\beta = 0.3$.

When comparing the convergence rate, the performance of aDTS with $\epsilon_{\mathrm{min}}^{\mathrm{Q2}}$ is noticeable lower especially on Rosenbrock's functions f_8, f_9 and Different powers f_{14} where the $\mathrm{err}_{\mathrm{RDE}}$ error often exceeds the lower error threshold even if a lower number of original-evaluated points would be sufficient for higher speedup of the CMA-ES. The adaptive control, on the other hand, helps especially on the Attractive sector f_6, which has the optimum in a point without continues derivatives and is therefore hard-to-regress by GPs, or on Shaffers' functions f_{17}, f_{18} where the aDTS is probably able to adapt to multimodal neighbourhood around function's optimum and performs best of all the compared algorithms. Within the budget of 250 FE/D, the aDTS (especially with $\epsilon_{\mathrm{min}}^{\mathrm{Q2}}$) is also able to find one of the best fitness value on regularly multimodal Rastrigin functions f_3, f_4 or f_{15} where the GP model apparently does not prevent the original CMA-ES from exploiting the global structure of a function.

Conclusion

Results of parameter tunning show that lower values of the exponential smoothing rate β provide better results. On the other hand, different combinations of slower and more rapid update behaviours bring better CMA-ES speedup for different kinds of functions, and choice of this parameter could depend on the experimenter's domain knowledge. We found that the adaptive approach speeds up the CMA-ES more than three other surrogate CMA-ES algorithms, namely DTS-CMA-ES, s*ACM-ES, and lmm-CMA, on several functions after roughly 150 FE/D.

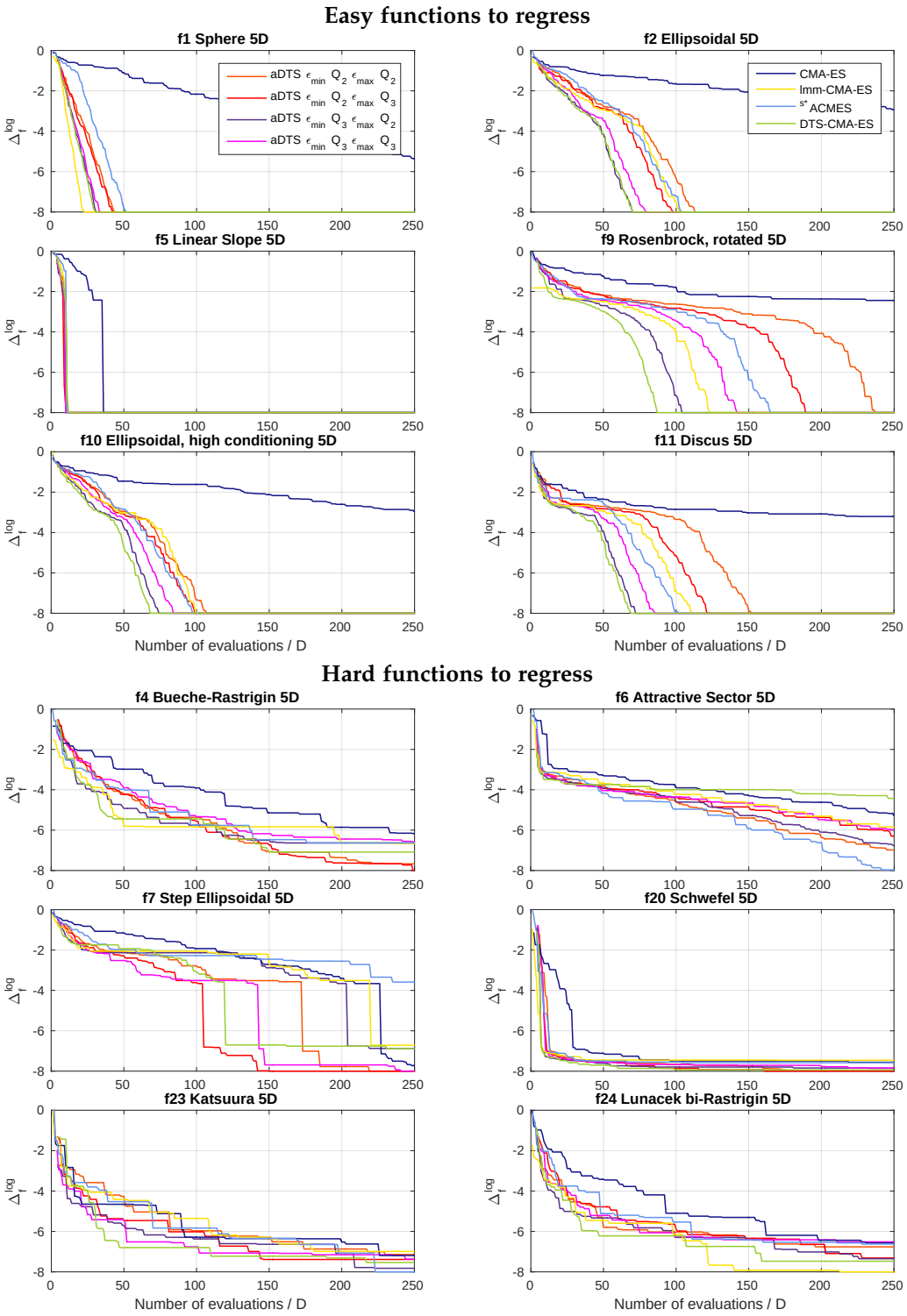


Figure 14: Algorithm comparison on 6 *easy* and 6 *hard* to regress COCO/BBOB noiseless functions in 5D. $\epsilon_{\min}$, $\epsilon_{\max}$: minimal and maximal error, $Q2$, $Q3$: median and third quartile.

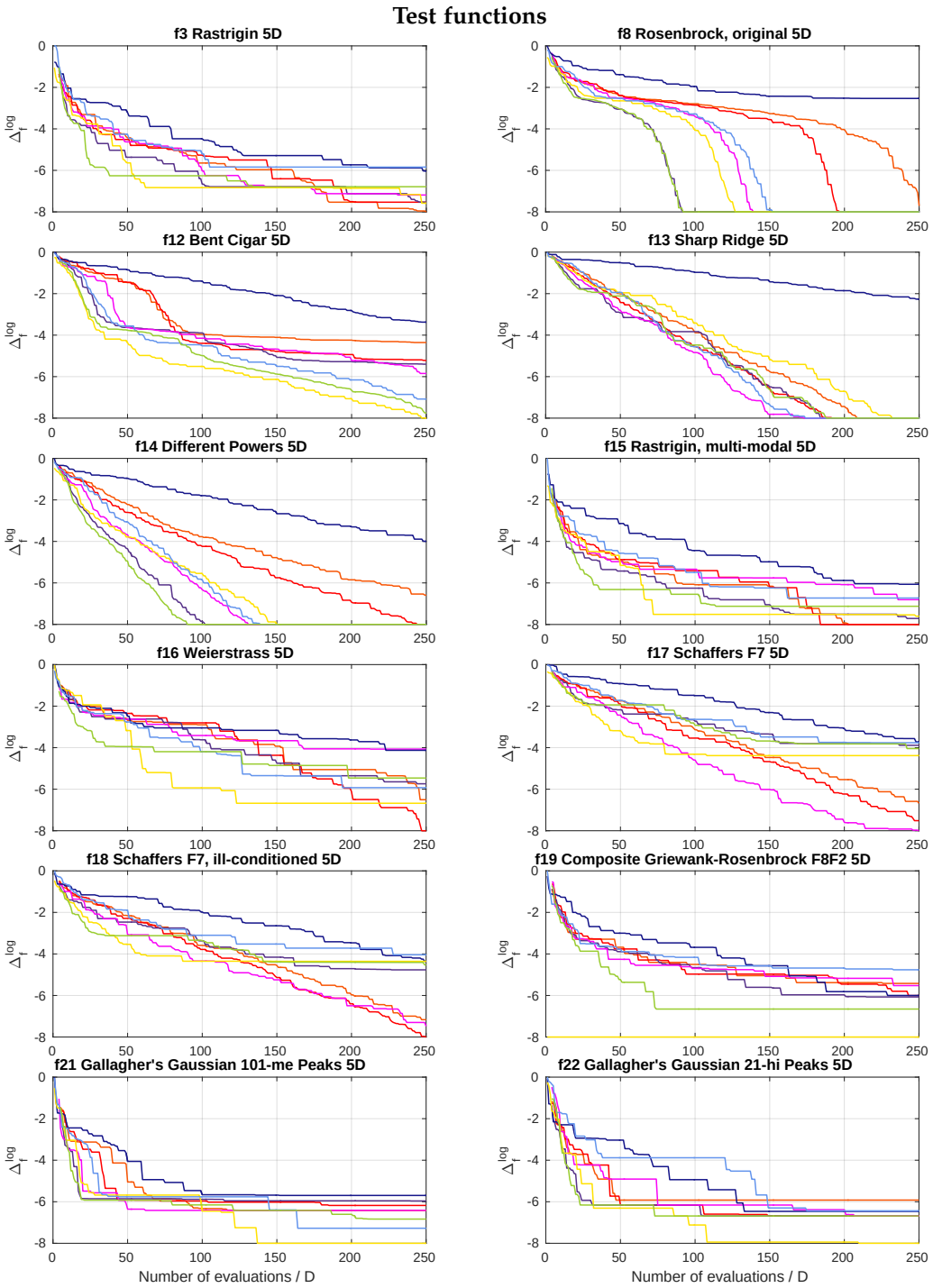


Figure 15: Algorithm comparison on 12 *test* function from the COCO/BBOB testbed in $5D$. $\epsilon_{\min}$, $\epsilon_{\max}$: minimal and maximal error, $Q2$, $Q3$: median and third quartile.

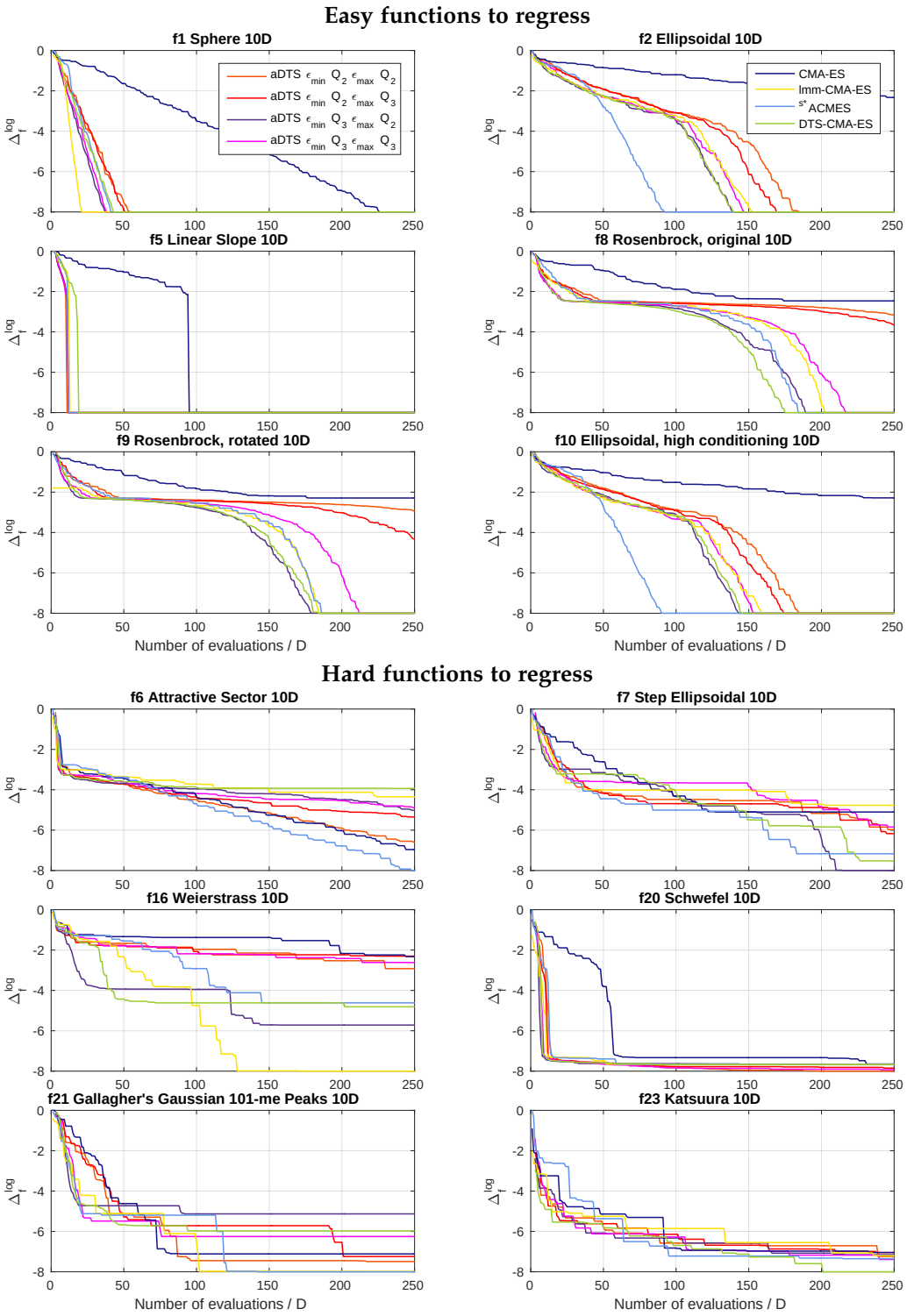


Figure 16: Algorithm comparison on 6 *easy* and 6 *hard* to regress COCO/BBOB noiseless functions in $10D$. $\epsilon_{\min}$, $\epsilon_{\max}$: minimal and maximal error, $Q2$, $Q3$: median and third quartile.

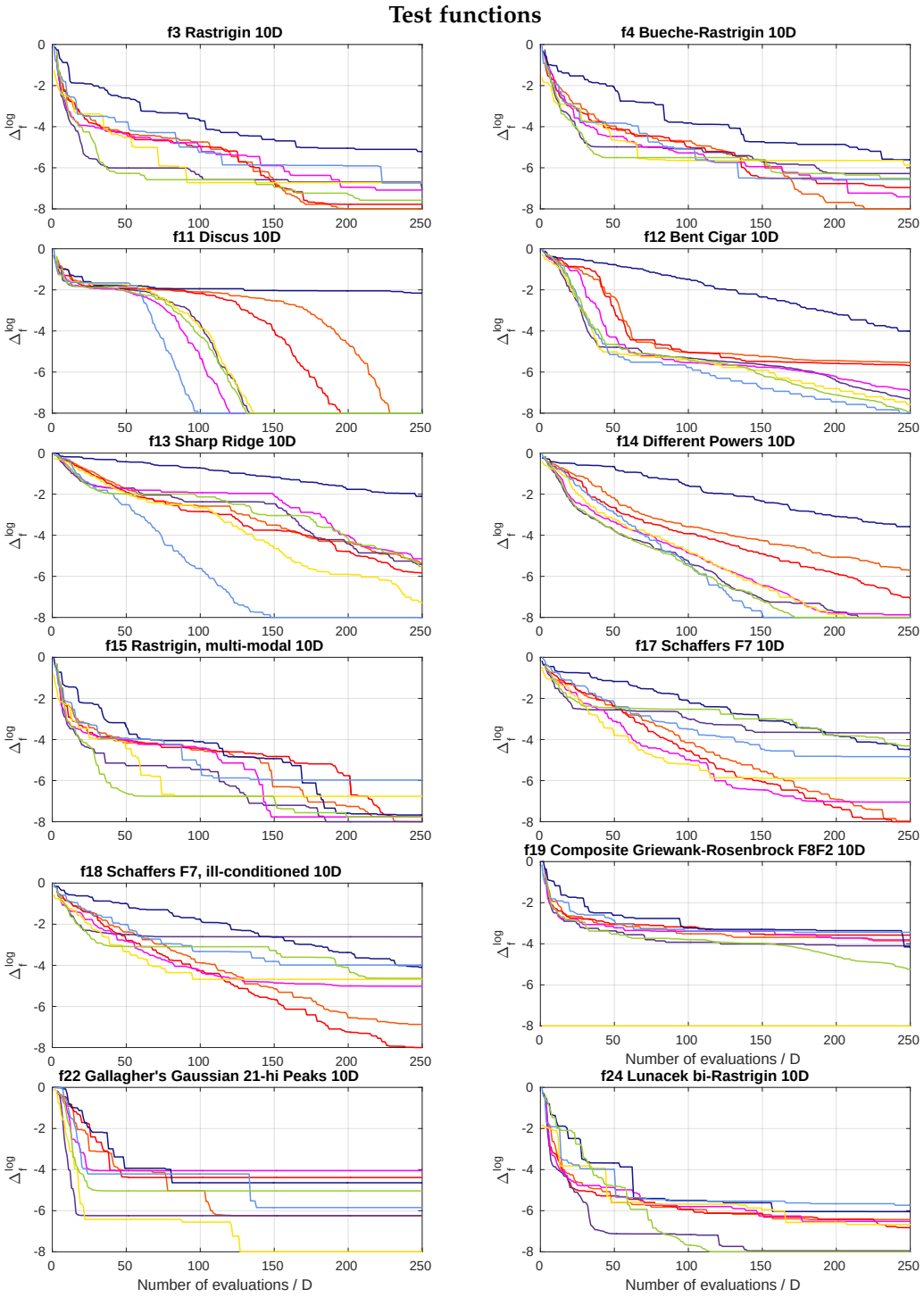


Figure 17: Algorithm comparison on 12 *test* function from the COCO/BBOB testbed in $10D$. $\epsilon_{\min}$, $\epsilon_{\max}$: minimal and maximal error, $Q2$, $Q3$: median and third quartile.

Table 8: The *easiest* (1.–6.) and the *hardest* (19.–24.) to regress six COCO/BBOB functions by the Gaussian process used in the DTS-CMA-ES (columns f) according to the corresponding medians of $\mathrm{err}_{\mathrm{RDE}}$ measured in 25 generations from 5 instances on independent testsets of size $\lambda = 8 + \lfloor 6 \log D \rfloor$ using $\mu = \lfloor \lambda/2 \rfloor$.

	2D		3D		5D		10D		20D	
	f	$\mathrm{err}_{\mathrm{RDE}}$	f	$\mathrm{err}_{\mathrm{RDE}}$	f	$\mathrm{err}_{\mathrm{RDE}}$	f	$\mathrm{err}_{\mathrm{RDE}}$	f	$\mathrm{err}_{\mathrm{RDE}}$
1.	5	0.00	5	0.00	5	0.00	5	0.04	5	0.04
2.	1	0.08	1	0.11	1	0.16	1	0.18	1	0.08
3.	2	0.10	2	0.13	10	0.18	10	0.26	24	0.18
4.	10	0.10	10	0.14	2	0.21	8	0.27	15	0.19
5.	11	0.10	9	0.16	11	0.23	2	0.27	19	0.20
6.	8	0.14	8	0.18	9	0.24	9	0.29	3	0.21
19.	18	0.46	23	0.43	24	0.41	21	0.40	18	0.38
20.	20	0.52	15	0.43	4	0.44	16	0.41	23	0.47
21.	24	0.53	24	0.49	23	0.45	23	0.47	6	0.48
22.	6	0.54	20	0.51	6	0.51	6	0.50	21	0.51
23.	7	0.54	6	0.52	20	0.54	20	0.54	20	0.52
24.	19	0.54	7	0.54	7	0.56	7	0.57	7	0.56

Table 9: Counts of the 1st ranks from 12 benchmark *test* functions from the BBOB/COCO testbed according to the lowest achieved Δ_f^{med} for different $\mathrm{FE}/D = \{25, 50, 100, 200\}$ and dimensions $D = \{2, 3, 5, 10\}$. Ties of the 1st ranks are counted for all respective algorithms. The ties often occure when $\Delta f_T = 10^{-8}$ is reached (mostly on f_1 and f_5).

	2D				3D				5D				10D				$\sum$			
FE/D	25	50	100	200	25	50	100	200	25	50	100	200	25	50	100	200	25	50	100	200
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.3$	1	0	0	2	1	2	2	3	0	0	0	0	0	0	0	1	2	2	2	6
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.4$	0	0	0	2	1	0	0	3	0	0	0	0	0	0	0	0	1	0	0	5
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.5$	2	0	0	3	0	0	0	3	0	0	0	0	0	0	0	1	2	0	0	7
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.3$	2	0	1	4	1	0	0	3	0	0	2	2	0	0	0	2	3	0	3	11
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.4$	0	0	1	4	0	0	0	4	0	1	2	3	0	0	**3**	**4**	0	1	6	15
$\epsilon^{Q2}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.5$	1	0	0	3	0	0	0	4	0	0	0	3	0	0	0	1	1	0	0	11
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.3$	1	0	**3**	4	**4**	**5**	**5**	**5**	**5**	3	**3**	**5**	**7**	**6**	**3**	**4**	**17**	**14**	**14**	**18**
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.4$	1	**4**	1	4	1	0	1	4	**5**	**4**	0	4	1	2	**3**	2	8	10	5	14
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q2}_{\mathrm{max}}, \beta = 0.5$	1	3	1	**5**	0	2	2	3	1	0	1	2	3	0	0	1	5	5	4	11
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.3$	0	2	2	4	1	3	3	3	0	3	**3**	4	0	1	2	1	1	9	10	12
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.4$	0	2	**3**	4	3	0	3	4	1	1	0	4	0	2	0	1	4	5	6	13
$\epsilon^{Q3}_{\mathrm{min}}, \epsilon^{Q3}_{\mathrm{max}}, \beta = 0.5$	**3**	1	2	3	0	0	1	3	0	0	1	2	1	1	1	2	4	2	5	10

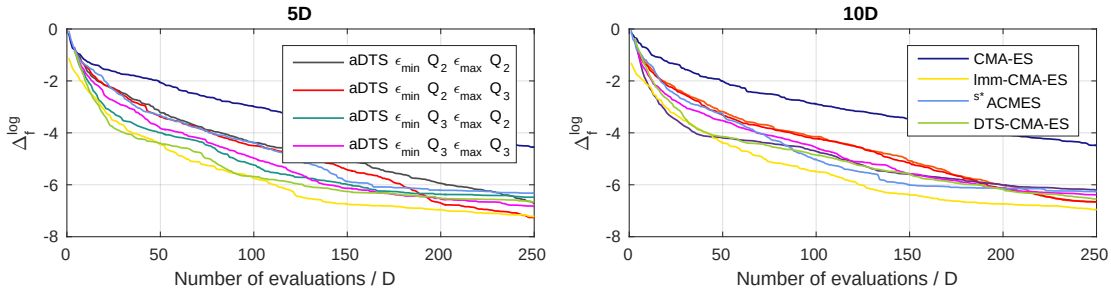


Figure 18: Algorithm comparison using averaged $\Delta_f^{\log}$ values on 12 *test* functions from the COCO/BBOB testbed in $5D$ and $10D$. $\epsilon_{\min}$, $\epsilon_{\max}$: minimal and maximal error, $Q2$, $Q3$: median and third quartile.

Table 10: A pairwise comparison of the algorithms on 12 *test* functions in $10D$ over the COCO/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 Bergmann-Hommel correction at family-wise significance level $\alpha = 0.05$.

10*D*	$\epsilon_{\min}^{Q2}$, $\epsilon_{\max}^{Q2}$		$\epsilon_{\min}^{Q2}$, $\epsilon_{\max}^{Q3}$		$\epsilon_{\min}^{Q3}$, $\epsilon_{\max}^{Q2}$		$\epsilon_{\min}^{Q3}$, $\epsilon_{\max}^{Q3}$		CMA-ES		lmm-CMA		^{s}*ACM-ES		DTS-CMA-ES	
#FEs/#FE$_T$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
$\epsilon_{\min}^{Q2}$, $\epsilon_{\max}^{Q2}$	—	—	6	5	4	6	4	8	12	11	2	5	6	8	3	7
$\epsilon_{\min}^{Q2}$, $\epsilon_{\max}^{Q3}$	6	7	—	—	3	5	5	6	11	10	2	6	7	7	2	5
$\epsilon_{\min}^{Q3}$, $\epsilon_{\max}^{Q2}$	8	6	9	7	—	—	7	7	12*	9	4	5	8	5	3	2
$\epsilon_{\min}^{Q3}$, $\epsilon_{\max}^{Q3}$	8	4	7	6	5	5	—	—	11*	10	4	6	9	7	4	4
CMA-ES	0	1	1	2	0	3	1	2	—	—	1	1	2	4	0	1
lmm-CMA	10	7	10	6	8	7	8	6	11*	11	—	—	10	6	7	5
^{s}*ACM-ES	6	4	5	5	4	7	3	5	10	8	2	6	—	—	3	7
DTS-CMA-ES	9	5	10	7	9	10	8	8	12*	11*	5	7	9	5	—	—

4.4.2 *Configuration of the Gaussian Process Models*

This section provides the implementation details of the training of and predicting with the Gaussian processes we have proposed in (Bajer et al., 2019) as well as the evaluation of their parameter settings.

Table 11: Bounds and starting values for hyperparameters MLE; $\mathbf{y}_N$ is a vector of the $\mathfrak{b}$-values from the GP training set and $\Delta_y = \max \mathbf{y}_N - \min \mathbf{y}_N$.

hyperparameter	starting value	lower bound	upper bound
m_μ	median $\mathfrak{b}$-value $\operatorname{med} \mathbf{y}_N$	$\min \mathbf{y}_N - 2\Delta_y$	$\max \mathbf{y}_N + 2\Delta_y$
σ_f^2	0.5	$\exp(-2)$	$\exp(25)$
ℓ	2	$\exp(-2)$	$\exp(25)$
σ_n^2	10^{-2}	10^{-6}	10

Table 12: Parameters of the GP surrogate models. The maximum distance $r_{\max}$ is derived using the Mahalanobis distance given by the covariance matrix $\sigma^2\mathbf{C}$. $Q_{\chi^2}(0.99, D)$ is the 0.99-quantile of the χ^2_D distribution, and therefore $\sqrt{Q_{\chi^2}(0.99, D)}$ is the 0.99-quantile of the norm of a D-dimensional normally distributed random vector.

parameter	considered values
training set selection method TSS	TSS closest, TSS cluster, TSS nearest, TSS recent
maximum distance $r_{\max}$	$2\sqrt{Q_{\chi^2}(0.99, D)}$, $4\sqrt{Q_{\chi^2}(0.99, D)}$
$N^{\mathcal{M}}_{\max}$	$10 \cdot D$, $15 \cdot D$, $20 \cdot D$
covariance function κ	κ_{SE}, $\kappa^{3/2}_{\text{Mat}}$, $\kappa^{5/2}_{\text{Mat}}$

The estimation of the GP model hyperparameters in Step 6 of the Algorithm 5 is performed using *maximum likelihood estimation* (MLE). It is started at and bounded by values obtained from brief experiments, see Table 11 for the exact values. The GP models employed in our algorithms make use of the GPML 4.0—a toolbox which accompanied the book of Rasmussen and Williams (2006). The only modifications are using the Matlab `fmincon` optimizer or the CMA-ES for the MLE of the GP hyperparameters and slight modifications in error handling in the case of numerical instabilities. The CMA-ES is used instead of `fmincon` if `fmincon` fails due to an infeasible starting point or numerical problems.

Evaluation of GP model parameters

Because we are not aware of any systematic comparison of different parameter settings of the GP surrogate models for the CMA-ES in literature, we provide such evaluation on the data from the COCO benchmark functions. Table 12 lists all the tested parameters with their considered values for four TSS methods using the $r_{\max}$ denotes the maximum distance between $\mathbf{m}^{(g)}$ and the points to be selected from $\mathcal{A}$. The minimum training set size $N^{\mathcal{M}}_{\min}$ was $3 \cdot D$ in all our GP models studied in (Bajer et al., 2019).

As training and testing datasets for this comparison, three generations from the first half and three generations from the second half of optimization runs of the DTS-CMA-ES ($\alpha = 0.05$), limited to 250 FE/D, were recorded on the first five instances of 24 noiseless COCO functions in dimensions 2, 3, 5, 10, and 20. For each recorded generation, the CMA-ES state variables, archive $\mathcal{A}$ and populations (as testing sets) were saved.

Figure 19 depicts the third quartiles of $\text{err}_{\text{RDE}}/r_{\text{succ}}$ where r_{succ} denotes the ratio of successful trainings of the second model $\mathcal{M}_2$. Both the errors and ratios were taken for each model setting of the full-factorial design of the parameters from Table 12 across 15 repetitions (5 instances

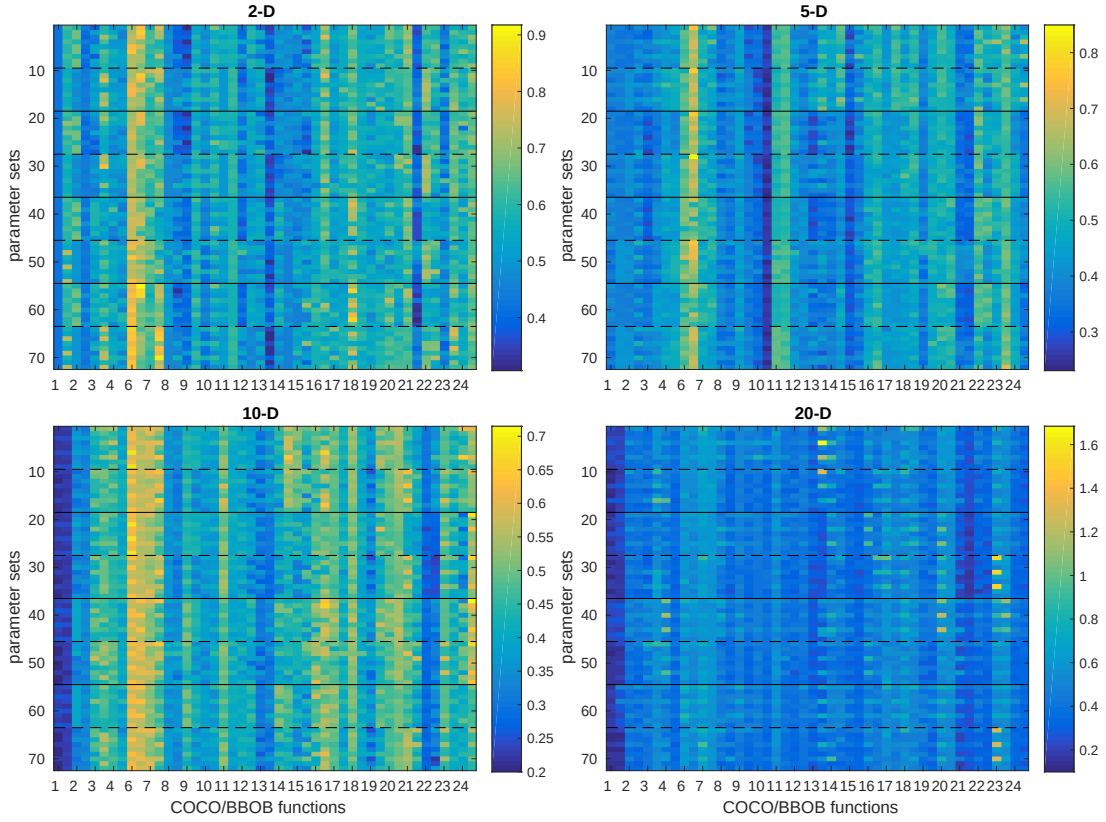


Figure 19: The third quartiles from sets of 15 values of $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$. Results for Linear slope f_5 (often equal to zero) were omitted for better visibility of other results. Two columns per each COCO function belong to the first and second half of optimization runs respectively. Solid horizontal lines separate different TSS methods in the order TSS recent, TSS nearest, TSS cluster, and TSS closest, dashed lines separate sectors with smaller and larger maximum distance $r_{\max}$. Three triples of settings within each sector represent raising values of $N^{\mathcal{M}}_{\max}$ and three covariance functions κ_{SE}, $\kappa^{3/2}_{\mathrm{Mat}}$, and $\kappa^{5/2}_{\mathrm{Mat}}$ within each triple.

$\times$ 3 generations) per each half of the optimization run on each function in each dimension. The same results concerning the influence of the individual parameters are summarized over functions via the n-way ANOVA in Table 13.

Not surprisingly, the highest error $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$ is visible on the Attractive sector f_6 function. This benchmark has its optimum in a point without continuous derivatives and with different slopes in different directions which is, therefore, very hard to regress by GPs. On the contrary, the lowest errors (in lower dimensions often equal to 0) were always achieved on the Linear slope f_5 whose results were omitted from the heatmaps for better color scaling of the remaining functions. Low errors were measured in higher dimensions on the Sphere function f_1; relatively high errors in lower dimensions on f_1 are probably due to a fast decrease of the CMA-ES step-size σ, which results in the low number of points in the training sets.

Let us look more closely at the effects of individual surrogate model parameters.

Table 13: GP surrogate model parameters. Results from the combination of n-way ANOVA analysis without interactions and the posthoc Tukey–Kramer's comparison tests of the effects of four parameters on the mean of $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$. The table shows the values of the parameters sorted according to the corresponding regressed $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$ (shown left) for the respective combinations of dimension and half of the optimization run. Values with the mean response according to Tukey–Kramer significantly higher than the respective lowest mean response are marked with ↓. Stars after the values sign the statistical significance of rejecting the ANOVA test hypotheses that the effects of the respective parameters on the mean responses of $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$ are negligible. Significant results of Tukey–Kramer's comparisons between the values of the respective parameters are marked in the gray lines of each block (1 = best value, 2 = second etc.). Stars $^{\star}/^{\star\star}/^{\star\star\star}$ sign the p-value lower than 0.05/0.01/0.001 for the respective tests.

dim	part of run	trainset selection	$r_{\max}$	$N^{\mathcal{M}}_{\max}$	covariance function
$2D$	i	0.40 TSS nearest 0.41 TSS recent 0.41 TSS cluster 0.41 TSS recent	0.41 2 0.41 4	0.40 $20 \cdot D$ 0.41 $15 \cdot D$ 0.41 $10 \cdot D$	0.40 $\kappa_{\mathrm{Mat}}^{5/2}$ 0.41 $\kappa_{\mathrm{Mat}}^{3/2}$ 0.41 κ_{SE}
$2D$	ii	0.42 TSS nearest 0.42 TSS recent 0.42 TSS cluster 0.42 TSS recent	0.42 $2^{\star\star}$ 0.42 4 ↓	0.42 $20 \cdot D$ 0.42 $15 \cdot D$ 0.42 $10 \cdot D$	0.42 κ_{SE} 0.42 $\kappa_{\mathrm{Mat}}^{5/2}$ 0.42 $\kappa_{\mathrm{Mat}}^{3/2}$
			$1 <^{\star\star} 2$		
$3D$	i	0.38 TSS recent 0.38 TSS nearest 0.38 TSS cluster 0.39 TSS recent	0.38 4 0.39 2	0.38 $15 \cdot D$ 0.38 $10 \cdot D$ 0.38 $20 \cdot D$	0.38 $\kappa_{\mathrm{Mat}}^{3/2}$ 0.38 $\kappa_{\mathrm{Mat}}^{5/2}$ 0.39 κ_{SE}
$3D$	ii	0.38 TSS recent*** 0.39 TSS nearest*** 0.39 TSS cluster ↓ 0.40 TSS recent ↓	0.39 2 0.39 4	0.39 $15 \cdot D$ 0.39 $10 \cdot D$ 0.39 $20 \cdot D$	0.39 $\kappa_{\mathrm{Mat}}^{3/2}$ 0.39 $\kappa_{\mathrm{Mat}}^{5/2}$ 0.40 κ_{SE}
		$1 <^{\star} 3$, $1 <^{\star\star\star} 4$, $2 <^{\star} 4$			
$5D$	i	0.34 TSS nearest*** 0.34 TSS recent*** 0.34 TSS cluster*** 0.35 TSS recent ↓	0.34 $2^{\star\star\star}$ 0.35 4 ↓	0.34 $10 \cdot D$ 0.34 $15 \cdot D$ 0.34 $20 \cdot D$	0.34 $\kappa_{\mathrm{Mat}}^{3/2\ \star\star\star}$ 0.34 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.35 κ_{SE}↓
		$1 <^{\star\star\star} 4$, $2 <^{\star} 4$, $3 <^{\star} 4$	$1 <^{\star\star\star} 2$		$1 <^{\star\star\star} 3$, $2 <^{\star\star\star} 3$
$5D$	ii	0.35 TSS nearest*** 0.35 TSS recent*** 0.35 TSS cluster*** 0.37 TSS recent ↓	0.35 $2^{\star\star}$ 0.36 4↓	0.35 $15 \cdot D$ 0.35 $10 \cdot D$ 0.35 $20 \cdot D$	0.35 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.35 $\kappa_{\mathrm{Mat}}^{3/2\ \star\star\star}$ 0.36 κ_{SE}↓
		$1 <^{\star\star\star} 4$, $2 <^{\star\star\star} 4$, $3 <^{\star\star} 4$	$1 <^{\star\star} 2$		$1 <^{\star\star} 3$, $2 <^{\star\star} 3$
$10D$	i	0.34 TSS nearest*** 0.34 TSS recent*** 0.34 TSS cluster ↓ 0.35 TSS recent ↓	0.33 $4^{\star\star\star}$ 0.35 2↓	0.34 $20 \cdot D$ 0.34 $15 \cdot D$ 0.34 $10 \cdot D$	0.33 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.34 κ_{SE}↓ 0.35 $\kappa_{\mathrm{Mat}}^{3/2}$↓
		$1 <^{\star} 3$, $1 <^{\star\star\star} 4$, $2 <^{\star\star} 4$	$1 <^{\star\star\star} 2$		$1 <^{\star\star} 2$, $1 <^{\star\star\star} 3$
$10D$	ii	0.32 TSS nearest*** 0.33 TSS recent ↓ 0.34 TSS cluster ↓ 0.34 TSS recent ↓	0.32 $4^{\star\star\star}$ 0.34 2↓	0.33 $20 \cdot D$ 0.33 $15 \cdot D$ 0.34 $10 \cdot D$	0.33 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.34 κ_{SE}↓ 0.34 $\kappa_{\mathrm{Mat}}^{3/2}$↓
		$1 <^{\star\star} 2$, $1 <^{\star\star\star} 3$, $1 <^{\star\star\star} 4$	$1 <^{\star\star\star} 2$		$1 <^{\star\star\star} 2$, $1 <^{\star\star\star} 3$
$20D$	i	0.35 TSS nearest*** 0.36 TSS recent ↓ 0.36 TSS cluster ↓ 0.36 TSS recent ↓	0.34 $4^{\star\star\star}$ 0.37 2↓	0.35 $20 \cdot D^{\star\star\star}$ 0.35 $15 \cdot D^{\star\star\star}$ 0.36 $10 \cdot D$ ↓	0.34 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.34 $\kappa_{\mathrm{Mat}}^{3/2}$↓ 0.39 κ_{SE}↓
		$1 <^{\star} 2$, $1 <^{\star\star\star} 3$, $1 <^{\star\star\star} 4$	$1 <^{\star\star\star} 2$	$1 <^{\star\star\star} 3$, $2 <^{\star\star} 3$	$1 <^{\star} 2$, $1 <^{\star\star\star} 3$, $2 <^{\star\star\star} 3$
$20D$	ii	0.34 TSS nearest*** 0.35 TSS recent ↓ 0.35 TSS recent ↓ 0.36 TSS cluster ↓	0.33 $4^{\star\star\star}$ 0.36 2↓	0.34 $20 \cdot D^{\star\star\star}$ 0.35 $15 \cdot D^{\star\star\star}$ 0.36 $10 \cdot D$ ↓	0.32 $\kappa_{\mathrm{Mat}}^{5/2\ \star\star\star}$ 0.34 $\kappa_{\mathrm{Mat}}^{3/2}$↓ 0.38 κ_{SE}↓
		$1 <^{\star} 2$, $1 <^{\star\star\star} 3$, $1 <^{\star\star\star} 4$, $2 <^{\star} 4$	$1 <^{\star\star\star} 2$	$1 <^{\star\star\star} 3$, $2 <^{\star\star\star} 3$	$1 <^{\star\star\star} 2$, $1 <^{\star\star\star} 3$, $2 <^{\star\star\star} 3$

- ◇ The dependence of the model error on the maximum radius of the training set $r_{\max}$ seems to be problem-dependent in low dimensions (in $2D$ and $5D$—lower errors can be observed for smaller radius, for example, on the Rastrigin's functions f_3 and f_{15} or on the multimodal Gallagher's Gaussians f_{21} and f_{22}). Rather surprising are the significantly lower errors for smaller radius in the $5D$, but Table 13 shows that larger radius yields significantly lower error in the $10D$ and $20D$ spaces.
- ◇ The model error appears not to depend on the number of training points $N^{\mathcal{M}}_{\max}$ (with the exception of $20D$ where both $20 \cdot D$ and $15 \cdot D$ points are significantly better than $10 \cdot D$ points). This insignificance is, however, affected by the number of points that are available for the training sets: the number of these points is often lower than $10 \cdot D$.
- ◇ Further, the choice of the covariance function has shown to be insignificant for $2D$ and $3D$, but starting from $5D$, the Matérn functions have shown to be significantly better and starting from $10D$ even more specifically with the parameter $\nu = 5/2$. Results in Figure 19 reveal that even though the squared exponential covariance exhibits in few cases the best performance (e. g., f_9 in $2D$), it is more often susceptible to training failure or high $\mathrm{err}_{\mathrm{RDE}}$, as can be seen, for example, on f_{21} and f_{23} in $10D$ or on several functions in $20D$.
- ◇ Finally, the TSS method of choice according to Table 13 is TSS nearest—taking the k nearest neighbors to each of the points in the population in a range $r_{\max}$, in lower dimensions closely followed by TSS recent—taking up to $N^{\mathcal{M}}_{\max}$ most recent points.

For the experiments in Section 4.4.3, the surrogate model settings performing significantly best in $10D$ and $20D$ were chosen for all dimensions: TSS = TSS nearest, $r_{\max} = 4\sqrt{Q_{\chi^2}(0.99, D)}$, $N^{\mathcal{M}}_{\max} = 20 \cdot D$, $\kappa = \kappa^{5/2}_{\mathrm{Mat}}$.

Fixed hyperparameter values

In addition to the evaluation of GP model parameters, we performed a brief investigation of inferring the values of two GP hyperparameters ℓ and m_μ from the DTS-CMA-ES state variables instead of their MLE.

Because the DTS-CMA-ES surrogate model already transforms the input coordinates into the space where the Euclidean distance equals to the Mahalanobis distance

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)\sigma^{-2}\mathbf{C}^{-1}(\mathbf{x}_1 - \mathbf{x}_2)}$$

in the original space, a natural value for the length-scale parameter ℓ would be for the squared-exponential covariance κ_{SE} (26) the constant $\ell = 1$ as it then corresponds to a simple exponential kernel

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sigma_f^2 \exp\left(-d(\mathbf{x}_1, \mathbf{x}_2)/2\right) \tag{86}$$

in the transformed space, proposed for RBF kernels by Loshchilov et al. (2010). Likewise, a common recommendation for the GP mean function is to use constantly zero $\mu(\mathbf{x}) = \mathbf{0}$ (Rasmussen and Williams, 2006). Note that the surrogate models standardize the $\mathfrak{b}$-values to zero mean regardless the mean function (step 5 in Algorithm 5).

Table 14 shows the third quartiles of $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$ for independently set $\ell = 1$ or $\mu(\mathbf{x}) = \mathbf{0}$ for two covariance functions, measured on the same datasets as in the previous evaluation—on the 2×15 populations of points per each COCO function and considered dimension. As

Table 14: Means and standard deviations of the third quartiles of $\mathrm{err}_{\mathrm{RDE}}/r_{\mathrm{succ}}$ for MLE and fixed hyperparameter values of the DTS-CMA-ES GP models. The third quartiles are from 2×15 independent datasets per function and dimension, and the means and standard deviations are averaged across 2×23 data (24 COCO functions without LINEAR SLOPE f_5 in the first and second half of optimization run) for each respective dimension. The higher the mean, the darker the color of the corresponding background is used.

cov. fun.	$\mu(\mathbf{x})$	ℓ	$2D$		$3D$		$5D$		$10D$		$20D$	
$\kappa_{\mathrm{Mat}}^{5/2}$	ML estimate	ML estimate	0.41	±0.07	**0.39**	±0.07	**0.33**	±0.06	**0.31**	±0.08	**0.31**	±0.12
$\kappa_{\mathrm{Mat}}^{5/2}$	ML estimate	$\ell = 1$	0.41	±0.06	0.39	±0.07	0.38	±0.07	0.42	±0.06	0.45	±0.08
$\kappa_{\mathrm{Mat}}^{5/2}$	$m_\mu = \mathbf{0}$	ML estimate	0.45	±0.08	0.42	±0.08	0.37	±0.09	0.33	±0.08	0.33	±0.11
$\kappa_{\mathrm{Mat}}^{5/2}$	$m_\mu = \mathbf{0}$	$\ell = 1$	**0.40**	±0.06	0.39	±0.07	0.39	±0.07	0.44	±0.06	0.49	±0.07
κ_{SE}	ML estimate	ML estimate	0.41	±0.07	0.39	±0.07	0.35	±0.05	0.32	±0.08	0.37	±0.13
κ_{SE}	ML estimate	$\ell = 1$	0.41	±0.06	0.40	±0.07	0.39	±0.07	0.42	±0.06	0.47	±0.09
κ_{SE}	$m_\mu = \mathbf{0}$	ML estimate	0.45	±0.09	0.44	±0.09	0.39	±0.08	0.36	±0.10	0.58	±0.34
κ_{SE}	$m_\mu = \mathbf{0}$	$\ell = 1$	0.41	±0.06	0.41	±0.07	0.40	±0.07	0.44	±0.06	0.51	±0.09

can be seen, using the fixed lengthscale $\ell = 1$ provides practically the same results as MLE ℓ for $D = 2, 3$, but is harmful in higher dimensions $10D$ and $20D$ where fitting the lengthscale seems to be essential for a good GP prediction. On the other hand, using the constant mean function $\mu(\mathbf{x}) = \mathbf{0}$ in connection with the Matérn covariance in higher dimensions does not dramatically deteriorate the prediction, but provides worse results in low-dimensional spaces.

Conclusion

In summary, the ability of Gaussian processes to predict the right ranking varies to a large extent with a problem at hand. On the contrary, a particular *type of covariance function* or some of the training set parameters, such as the algorithm of *selection points from the archive* or the *maximum radius* for this selection, might be preferred, especially in higher dimensions.

4.4.3 *Evaluation of Gaussian Process Based CMA-ES Algorithms*

In (Bajer et al., 2019), we have thoroughly compared the DTS-CMA-ES, most of GP-assisted CMA-ES versions with each other as well as with a few other important black-box optimization methods on the COCO benchmarks. All the algorithms and experiments proposed by the authors are implemented in Matlab[5].

DTS-CMA-ES Implementation and Parameter Settings

The implementation of the DTS-CMA-ES is based on the IPOP-CMA-ES in the Matlab version 3.62β. For the COCO benchmarks, the initial mean is uniformly sampled from $[-4, 4]^D$, the

5 The source code is available on Github https://github.com/bajeluk/surrogate-cmaes

initial step-size $\sigma^{(0)} = \frac{8}{3}$ and the number of IPOP restarts is limited to 50. Changes in the CMA-ES code are rather subtle: the doubly trained EC is employed just after the CMA-ES new population sampling (step 3 in Algorithm 9), and the EC returns to the CMA-ES a mix of the original and GP model fitness, scaled such that none of the predicted fitness is lower than the so-far minimum original $\mathfrak{b}$-value in $\mathcal{A}$.

Both the GP model trainings in the DTS-CMA-ES (steps 4 and 9) can fail, usually either due to numerical errors in Cholesky matrix decomposition during GP hyperparameter estimation (used for the GP covariance matrix inversion), or the model is over-fitted in such a way that it returns a constant almost everywhere. If training the first model $\mathcal{M}_1$ fails and there is no successfully trained model at most two generations old, the algorithm proceeds as the standard CMA-ES: the whole population is evaluated with the original fitness. If training the second model $\mathcal{M}_2$ fails, the first model is used for the final prediction, too, and the self-adaptation step is skipped.

RATIO OF ORIGINAL-EVALUATED POINTS α The ratio of original-evaluated points should inversely depend on the model error and can be either set fixed or can be adapted by the DTS-CMA-ES. For the evaluations budget of roughly $50 - 200\,\mathrm{FE}/D$, the experimentally discovered fixed value $\alpha = 0.05$ has shown to achieve one of the best results on the COCO testbed ((Pitra et al., 2016), updated results are provided later in this section). The self-adaptation setting is, on the other hand, better for larger evaluation budgets and for harder-to-regress functions, and its setting is described below.

POPULATION SIZE Replacing the exact fitness values with values returned by a surrogate model introduces an uncertainty of which the CMA-ES is not aware. The CMA-ES updates are then too rapid with respect to the number of original FE/D, and especially the step-size $\sigma^{(g)}$ is shrunk too fast. As a result, the CMA-ES is often trapped in a local optimum. As the goal of our work is not a precise fine-tuning of the CMA-ES internal constants, we use the population size $\lambda = 8 + \lfloor 6 \ln D \rfloor$ that is doubled compared to the default. This enlargement results in adequate CMA-ES updates and the convergence rate of the DTS-CMA-ES is considerably improved (Pitra et al., 2016).

CRITERIA FOR THE SELECTION OF ORIGINAL-EVALUATED POINTS Figure 20 shows aggregated results of all five mentioned criteria, separately for selected unimodal and multimodal COCO benchmarks. These graphs confirm both theoretical expectations and the previous investigations, e. g., in (Ulmer et al., 2003). Selecting the original-evaluated points based on the best GP predictive mean ($\mathfrak{C}_\mathrm{M}$) works most *exploitative*: it leads the algorithm towards the nearest local optimum. It thus works well on the unimodal functions, but it provides the worst results on the multimodal benchmarks.

Similarly, the $\mathfrak{C}_\mathrm{EI}$ is in comparison with the $\mathfrak{C}_\mathrm{PoI}$ less localized and thus more *explorative*. The $\mathfrak{C}_\mathrm{PoI}$ performs considerably better on the unimodal functions while the $\mathfrak{C}_\mathrm{EI}$ and also the $\mathfrak{C}_\mathrm{STD}$ have superior results on the multimodal functions where the $\mathfrak{C}_\mathrm{PoI}$ still performs adequately. $\mathfrak{C}_\mathrm{ERDE}$ has been shown neither the best nor the worst; it performs similarly to or slightly better than the $\mathfrak{C}_\mathrm{STD}$ on the unimodal functions, but considerably worse in the case of the multimodal benchmarks.

Altogether, the best average results on the whole COCO benchmark set is provided by the $\mathfrak{C}_\mathrm{PoI}$, which is used with the threshold $T = \mathfrak{b}_\mathrm{min} - 0.05(\mathfrak{b}_\mathrm{max} - \mathfrak{b}_\mathrm{min})$ in the rest of this work.

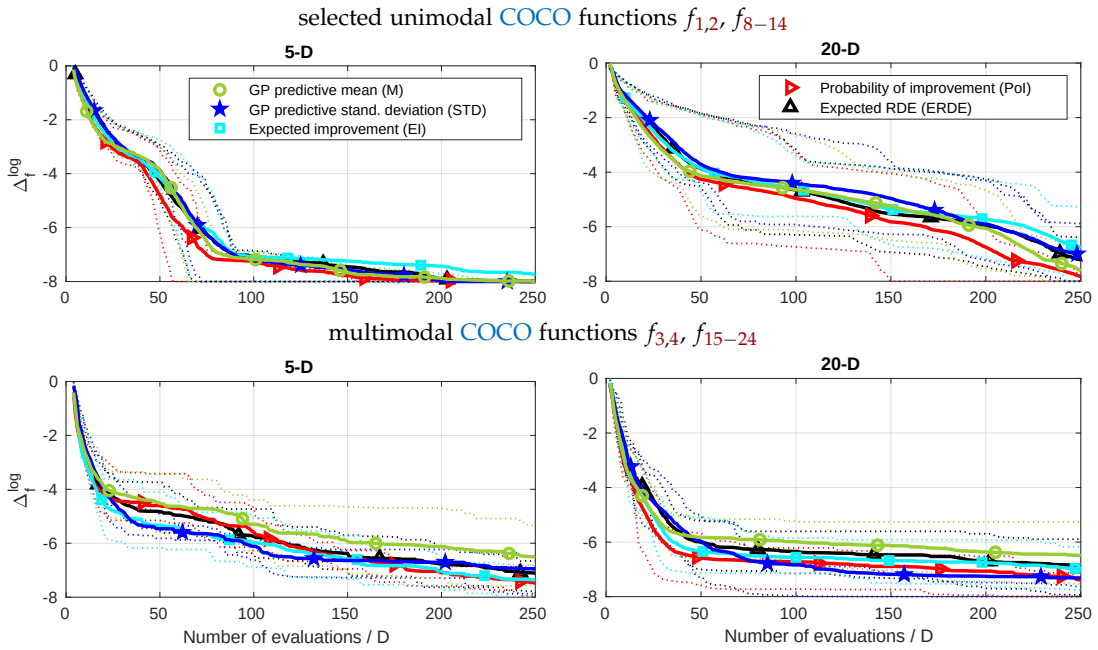


Figure 20: Criteria for the selection of original-evaluated points ($\mathfrak{C}_{\mathrm{M}}$, $\mathfrak{C}_{\mathrm{STD}}$, $\mathfrak{C}_{\mathrm{EI}}$, $\mathfrak{C}_{\mathrm{PoI}}$, $\mathfrak{C}_{\mathrm{ERDE}}$) in the $\alpha = 0.05$/DTS-CMA-ES. The $\log_{10}$ of the median best $\mathfrak{b}$-value distances to the optima were linearly scaled to $[-8, 0]$ and *averaged* (solid lines) over the selected unimodal (first row) and multimodal (second row) benchmarks separately. The first and third *quartiles* from the respective sets of medians are denoted with dotted lines.

However, it can be replaced with the $\mathfrak{C}_{\mathrm{EI}}$ or $\mathfrak{C}_{\mathrm{STD}}$ when some a priori knowledge about the multimodality of the fitness is available.

SELF-ADAPTATION PARAMETER SETTING The specific behavior of the proposed α-self-adaptation is determined by the following set of parameters whose values we have investigated in (Pitra et al., 2017a) – see Section 4.4.1.

- β – exponential smoothing update rate (a. k. a. learning rate) moderates vivid α oscillations. In Pitra et al. (2017a), we recommend $\beta = 0.3$, which is used in the following experiments, too, and which performs very similarly to the value $\beta = 0.2$, the value used in the ^{s*}ACM-ES (Loshchilov et al., 2013b). The DTS-CMA-ES has shown to be rather robust with respect to the exact setting of this parameter.

- $\alpha_{\min}$, $\alpha_{\max}$ – minimum and maximum bounds for the ratio α—should be set to enable the self-adaptation to adequately react on a broad set of problems. The used minimum bound $\alpha_{\min} = 0.04$ corresponds to one evaluated point per generation in $D = 2, \ldots, 20$, and the maximum bound should be set to $\alpha_{\max} = 1.0$ for the possibility of disabling the surrogate modeling completely on the fitness landscapes where the GPs are not able to satisfyingly predict the population ranking.

- ◇ $\epsilon_{\min}$, $\epsilon_{\max}$ – lower and upper saturation bounds on the exponentially smoothed $\mathrm{err}_{\mathrm{RDE}}$ error for the linear transfer function. These parameters determine the self-adaptive behavior to a large extent and should change with both the dimension D of the problem and the actual used ratio $\alpha^{(g+1)}$. We have used the proposed set of multiple linear regression models $\epsilon_{\min}^{\mathrm{Q2}}$ and $\epsilon_{\max}^{\mathrm{Q3}}$ from Equations (78) and (81), respectively, which are most capable of adapting to changing GP ranking error. Because the calculation of the values $\epsilon_{\min}$, $\epsilon_{\max}$ and the calculation of the resulting ratio $\alpha^{(g+1)}$ (step 3 in Algorithm 10) are mutually dependent, they are alternately repeated in a cycle until convergence or 500 iterations.

Experimental Settings of Compared Algorithms

This section comprises the settings of the algorithms we assessed in (Bajer et al., 2019). The six algorithms based on the CMA-ES and a GP model: the S-CMA-ES, the DTS-CMA-ES in both the fixed and self-adaptive version, the MA-ES, GPOP, and SAPEO. For the sake of comparison with other state-of-the-art black-box optimizers, we took into considerations the results of the standard IPOP-CMA-ES with both the recommended and the doubled population size, two other surrogate-assisted CMA-ES algorithms lmm-CMA and ^{s*}ACM-ES, the SMAC algorithm as a representative of Bayesian optimization algorithms, and two local-search numerical optimization algorithms based on a trust region method: the BOBYQA algorithm by Powell (2009) and the interior-point method (Byrd et al., 2000) from the Matlab `fmincon` function.

Whereas the detailed parameter setting of the DTS-CMA-ES can be found above, the list of the settings of the remaining algorithms follows.

S-CMA-ES: population size $\lambda = 4 + \lfloor 3 \ln D \rfloor$, model life-length $g_{\mathcal{M}} = 5$, Gaussian process surrogate model: TSS cluster, $r_{\max} = 10$, $N_{\max} = 15 \cdot D$, $\kappa_{\mathrm{Mat}}^{5/2}$.

MA-ES: population size $\lambda = 10$, parent number $\mu = 2$, number of training points 2λ, extended population size $\lambda_{\mathrm{Pre}} = 3\lambda$, predictive mean as pre-selection criterion, our Matlab implementation based on the implementation of the S-CMA-ES.

GPOP: population size $\lambda = 10$, parent number $\mu = 2$, training set size parameters $N_C = N_R = 5D$, input space termination tolerance 10^{-8}, minimum fitness change for 10 last iterations unless restart 10^{-9}, perturbation size $m = 1$, our GPML-based Matlab implementation using the Gaussian processes from the S-CMA-ES.

SAPEO: the COCO/BBOB results have been kindly provided by the algorithm's authors; variant *ucp-2*: sample-size 2λ, using three increasingly risky dominance relations.

CMA-ES: the IPOP-CMA-ES version (Matlab code 3.62β) with single $(4 + \lfloor 3 \ln D \rfloor)$ and doubled $(8 + \lfloor 6 \ln D \rfloor)$ population size, 50 IPOP restarts, $\sigma^{(0)} = \frac{8}{3}$, other settings were left default.

LMM-CMA, ^{s*}ACM-ES, SMAC: results downloaded from the COCO/BBOB results archive, the BIPOP-^{s*}ACM-ES-k by Loshchilov et al. (2013a) used for the ^{s*}ACM-ES, SMAC results are only up to 100 FE/D; initializations of the lmm-CMA and SMAC are rather disputable on f_{19}.

BOBYQA, `fmincon`: both algorithms initialized from a new uniformly sampled point every fifth restart, otherwise used in their default settings, Dlib C++ implementation of BOBYQA, `fmincon` with the interior-point algorithm from Matlab 2017a.

Validation results

This section assesses the performance of all the algorithms mentioned in the previous section. The performances are presented by the graphs in Figures 21–25[6]. The numbers of COCO functions for which one algorithm is better than each of the others are presented in Table 15. Table 16 shows average ranks of the individual algorithms over all COCO functions and the Iman-Davenport statistic F_F with the corresponding p-value of the test where the null hypothesis would mean equally distributed performance of all algorithms. Not surprisingly, the test rejects this hypothesis in both $5D$ and $20D$.

The graphs in Figures 21–24 document the differences between the GP/CMA-ES algorithms. They depict the best-achieved logarithmic median distances to the benchmarks' optima $\Delta_f^{\log}$ for the respective numbers of FE/D. These medians (and in the case of the GP/CMA-ES algorithms also the first and third quartiles) are from 15 independent instances for each respective algorithm, function and dimension. The results of the GP/CMA-ES algorithms are represented by thick lines and the reference algorithms use thinner lines to see whether some better algorithm than any of the GP/CMA-ES exists (see Section 2.6.2 for detailed explanation of convergence graphs).

In addition, comparison of the same set of 13 algorithms, based on the COCO-provided ECDF graphs (see Section 2.6.1), is depicted in Figures 26–28. Algorithm performance for low numbers of FE/D is emphasized due to the logarithmic scale of the horizontal axis, and the results for $10D$ are included, too.

S-CMA-ES A noticeable pattern among our algorithms is, with few exceptions in $20D$, that the S-CMA-ES almost always converges to the optimum at the same rate or slower than both versions of DTS-CMA-ES. It might be at least partly because the generation-based evolution control of the S-CMA-ES is not able to exploit the GP models uncertainty, and, therefore, we will point our attention to the DTS-CMA-ES.

DTS-CMA-ES The fixed ($\alpha = 0.05$) version of the DTS-CMA-ES converges fastest out of the six GP/CMA-ES algorithms particularly on the subset f_{8-14} of the unimodal functions (which are more or less after some transformation similar to the Rosenbrock's function f_8) and also on the highly multimodal functions $f_{19-21,23,24}$, often functions without any global structure. While some of the non-GP algorithms ^{s*}ACM-ES, lmm-CMA, BOBYQA or `fmincon` perform similarly or better on the first part of functions (f_{8-14}), the performance on the multimodal functions ($f_{19-21,23,24}$) is often the best out of all compared algorithms. These results are probably because the fixed DTS-CMA-ES is able to descend to the functions' optima regardless of the relatively high model error which forces the adaptive algorithms to spend more original evaluations than necessary. On the other hand, the limitations of GP models can be seen on the non-smooth functions f_6 or f_{13} where the GP is not able to regress the fitness around the optimum and where other algorithms perform better.

The self-adaptive DTS-CMA-ES is mostly the second or the third GP/CMA-ES algorithm on the functions where the fixed DTS-CMA-ES succeeds (see the previous paragraph). However, the self-adaptation improves the fixed DTS-CMA-ES behavior on the multimodal functions with a global structure, especially on the Rastrigin functions $f_{3,4,15}$ and Shaffers functions $f_{17,18}$ where the self-adaptive DTS-CMA-ES is usually the best of all 13 algorithms. A closer

6 Additional results are on the author's webpage `http://bajeluk.matfyz.cz/gpcomparison2017`

look at the convergence graphs reveals that the self-adaptive DTS-CMA-ES indeed speeds up the 2-population IPOP-CMA-ES on these functions, approximately by the factor of 2.

MA-ES Our implementation of the MA-ES most often occupies the worse half of the GP-based algorithm ranking, even on the unimodal functions for which the $(2, 10)$ setting was, according to its authors, mainly aimed. The results are often very similar to or slightly better than the results of the IPOP-CMA-ES, which signifies that the proposed individual-based EC is not able to adequately speed up the CMA-ES compared to the EC of the other surrogate algorithms. However, the results on the ATTRACTIVE SECTOR f_6 are rather surprising because the MA-ES' model does not mislead the search even though the function is not smooth in the optimum and thus very hard to regress by GPs.

GPOP, SMAC Although the GPOP belongs to Bayesian optimization algorithms, it descends much closer to the global optima than SMAC. For very few FE/D, SMAC is slightly faster but very soon traps in some local optima. The GPOP is often the slowest or the second slowest GP/CMA-ES algorithm on the noiseless COCO set (in 20D on $f_{3,4}$, $f_{6,7}$, f_{10-16}, f_{19-24}), but it is the fastest during initial generations on the SPHERE f_1 and ELLIPSOIDAL f_2, where the GP is able to regress the function well even with a very small number of training points. The GPOP is the second on the SCHAFFERS' functions $f_{17,18}$ (rather surprisingly only in 20D, it is the slowest in 5D).

^{s*}ACM-ES The BIPOP-^{s*}ACM-ES' convergence speed is often similar or slightly worse than the self-adaptive DTS-CMA-ES in 5D. In 20D, however, the ^{s*}ACM-ES converges much faster compared to the other algorithms, and it is the best or the second among all 13 algorithms on the unimodal functions f_2, $f_{7,8}$, f_{10-14}. It probably profits from its ranking surrogate model and, therefore, from the invariance to monotonous transformations of the fitness. A considerably faster convergence is clearly visible especially on the ill-conditioned or non-smooth unimodal functions.

SAPEO The SAPEO algorithm usually converges at a similar rate as the MA-ES or the IPOP-CMA-ES. Nonetheless, its performance is considerably better than the MA-ES on functions f_9, f_{11}, f_{14}, f_{19}, f_{22}, and it is the best algorithm on f_6 and f_{16} in 20D, where it obviously adapts well to the fitness. Hence, the concept of uncertainty propagation is rather promising, especially considering the GP model was trained only on 2λ points.

LMM-CMA The quadratic-model-based lmm-CMA ranks on most benchmarks somewhere between or very close to the fixed and self-adaptive DTS-CMA-ES (f_{2-4}, f_{8-12}, f_{14-18}, $f_{21,22}$) which conforms with the average results in Figure 25 where it is very close to the self-adaptive DTS-CMA-ES. This similarity in terms of the results is in accordance with the fact that both algorithms use a metric (i. e., not a rank-based) surrogate model as well as a similar kind of adaptivity.

BOBYQA AND `fmincon` The BOBYQA is the best optimizer (out of all 13 compared) in the very beginning of the optimization, i. e., up to roughly $15 - 20\,\mathrm{FE}/D$. Later, the behavior of both the BOBYQA and `fmincon` dramatically changes with the benchmark at hand. They are rather successful on the unimodal functions—BOBYQA on f_1, f_5, $f_{8,9}$, and `fmincon` on the ill-conditioned f_{10-14} and also on $f_{1,2}$ and f_6 in 5D. Moreover, they are in few cases able to

Table 15: A pairwise comparison of the algorithms in 5D and 20D in terms of the number of wins of the row algorithm against the column algorithm over all noiseless COCO functions for two evaluations budgets. The budget (a) "1" denotes the smallest number of FE/D at which any of the algorithms reached the target $\mathbb{b}_t = \mathbb{b}_{opt} + 10^{-8}$ on the respective function, or 250 FE/D if the target was not reached, and (b) "1/3" is one third of the budget "1". The asterisk marks the row algorithm achieving a significantly lower value of the objective function than the column algorithm (on medians over 15 instances from 24 functions) according to the Friedman post-hoc test with the Shaffer correction at the family-wise level 0.05 (Demšar, 2006; García and Herrera, 2008).

5D	S-CMA-ES		0.05 DTS		adp. DTS		MA-ES		GPOP		SAPEO		CMA-ES		CMA-ES 2pop		^{s*}ACM-ES		lmm-CMA		BOBYQ		SMAC		fmincon	
eval. budget	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
S-CMA-ES	—	—	3	2	5	2	12	9	9	13	9	7	14	12	13	14	11	10	5	2	8	8	16	18	12	10
0.05 DTS	21*	22*	—	—	16	15	18	19*	21	23*	19	21*	19*	19*	21*	18*	20*	19*	15	16	16	19	19*	20*	16	22*
adp. DTS	19	22*	6	7	—	—	17	20	15	20*	18	19	22	23*	23*	23*	22	18	12	16	12	15	18	20*	13	15
MA-ES	12	15	6	5	7	4	—	—	12	15	13	12	18	16	20	15	16	11	8	6	10	10	14	19	11	10
GPOP	15	11	3	1	9	4	12	9	—	—	11	9	14	13	16	11	15	9	8	7	10	7	12	13	14	10
SAPEO	15	17	5	3	6	5	11	12	13	15	—	—	15	16	21	16	14	11	6	6	10	10	14	18	13	9
CMA-ES	10	12	5	5	2	1	6	8	10	11	9	8	—	—	18	10	11	11	3	6	11	9	12	15	11	9
CMA-ES 2pop	11	10	3	6	1	1	4	9	8	13	3	8	6	14	—	—	9	10	5	8	8	10	8	15	11	10
^{s*}ACM-ES	13	14	4	5	2	6	8	13	9	15	10	13	13	13	15	14	—	—	6	5	9	12	13	17	12	13
lmm-CMA	19	22	9	8	12	8	16	18	16	17	18	18	21	18	19*	16	18	19	—	—	14	14	17	19*	13	13
BOBYQA	16	16	8	4	12	8	14	14	14	17	14	14	13	15	16	14	15	12	10	10	—	—	15	17	16	12
SMAC	8	6	5	3	6	3	10	5	12	11	10	6	12	9	16	9	11	7	6	4	9	7	—	—	13	11
fmincon	12	14	8	2	11	9	13	14	10	14	11	15	13	15	13	14	12	11	11	11	8	12	11	13	—	—

20D	S-CMA-ES		0.05 DTS		adp. DTS		MA-ES		GPOP		SAPEO		CMA-ES		CMA-ES 2pop		^{s*}ACM-ES		lmm-CMA		BOBYQ		SMAC		fmincon	
eval. budget	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
S-CMA-ES	—	—	1	3	8	6	11	10	15	12	8	8	11	11	12	7	9	5	11	4	11	11	18	16	13	10
0.05 DTS	23	21	—	—	13	14	18	17	18*	18	17	16	19*	18	17*	14	15	6	17	14	19	17	20*	19*	17	13
adp. DTS	16	18*	9	10	—	—	20	21	20*	21*	17	18	22*	21	23*	21	15	9	17	12	17	13	20*	20*	16	11
MA-ES	13	14	6	7	4	3	—	—	15	15	8	3	18	11	17	9	8	4	6	6	12	11	16	18	12	11
GPOP	9	12	6	6	4	3	9	9	—	—	9	7	11	10	13	7	6	5	7	6	9	6	16	17	14	9
SAPEO	16	16	7	8	7	6	16	21	15	17	—	—	21	21	20	16	11	11	9	10	15	12	20	20*	16	11
CMA-ES	13	13	5	6	2	3	6	13	13	14	3	3	—	—	19	12	6	9	4	6	11	11	14	17	12	9
CMA-ES 2pop	12	17	7	10	1	3	7	15	11	17	4	8	5	12	—	—	5	8	5	10	8	13	14	19	12	9
^{s*}ACM-ES	15	19*	9	18	9	15	16	20	18	19*	13	13	18	15	19	16	—	—	10	13	16	17	17	21*	16	15
lmm-CMA	13	20	7	10	7	12	18	18	17	18	15	14	20	18	19	14	14	11	—	—	18	18	17*	20*	13	10
BOBYQA	13	13	5	7	7	11	12	13	15	18	9	12	13	13	16	11	8	7	6	6	—	—	15	17	13	13
SMAC	6	8	4	5	4	4	8	6	8	7	4	4	10	7	10	5	7	3	6	3	9	7	—	—	11	10
fmincon	11	14	7	11	8	13	12	13	10	15	8	13	12	15	12	15	8	9	11	14	11	11	13	14	—	—

successfully solve the multimodal benchmarks—BOBYQA adequately f_{19-24}, and `fmincon` is one of the best in 20D on $f_{21,22}$. But not surprisingly, these algorithms otherwise often stuck in local optima and on several benchmarks rank worse than the standard IPOP-CMA-ES.

Although it is not the prime criterion for the comparison of black-box optimizers, we also provide the average CPU time per one original function evaluation for the algorithms that we have benchmarked in Table 17.

4.4.4 *Conclusion*

This section re-investigates the employment of Gaussian processes in the CMA-ES. The presented algorithm DTS-CMA-ES differs from the lmm-CMA and ^{s*}ACM-ES not only in the regression model, but, more importantly, it uses Gaussian process uncertainty prediction for selecting points in its evolution control. The selection that does not use uncertainty performs

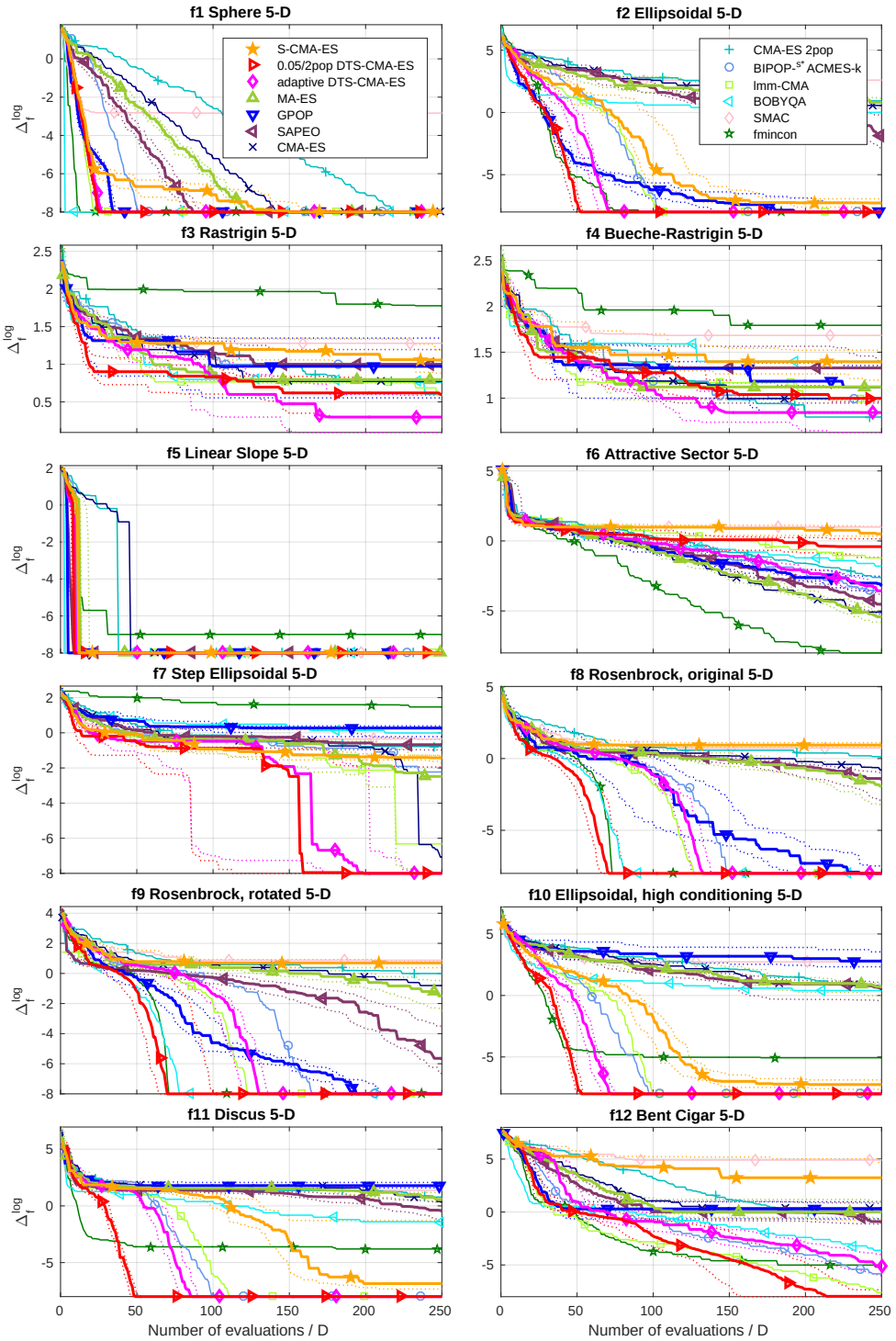


Figure 21: Medians (solid) and 1st/3rd quartiles (dotted) of the distances to the optima of 12 noiseless COCO benchmarks f_{1-12} in $5D$ for 13 black-box optimizers with the emphasis on the six Gaussian process/CMA-ES algorithms. Medians/quartiles were calculated across 15 independent instances for each algorithm and are shown in the $\log_{10}$ scale.

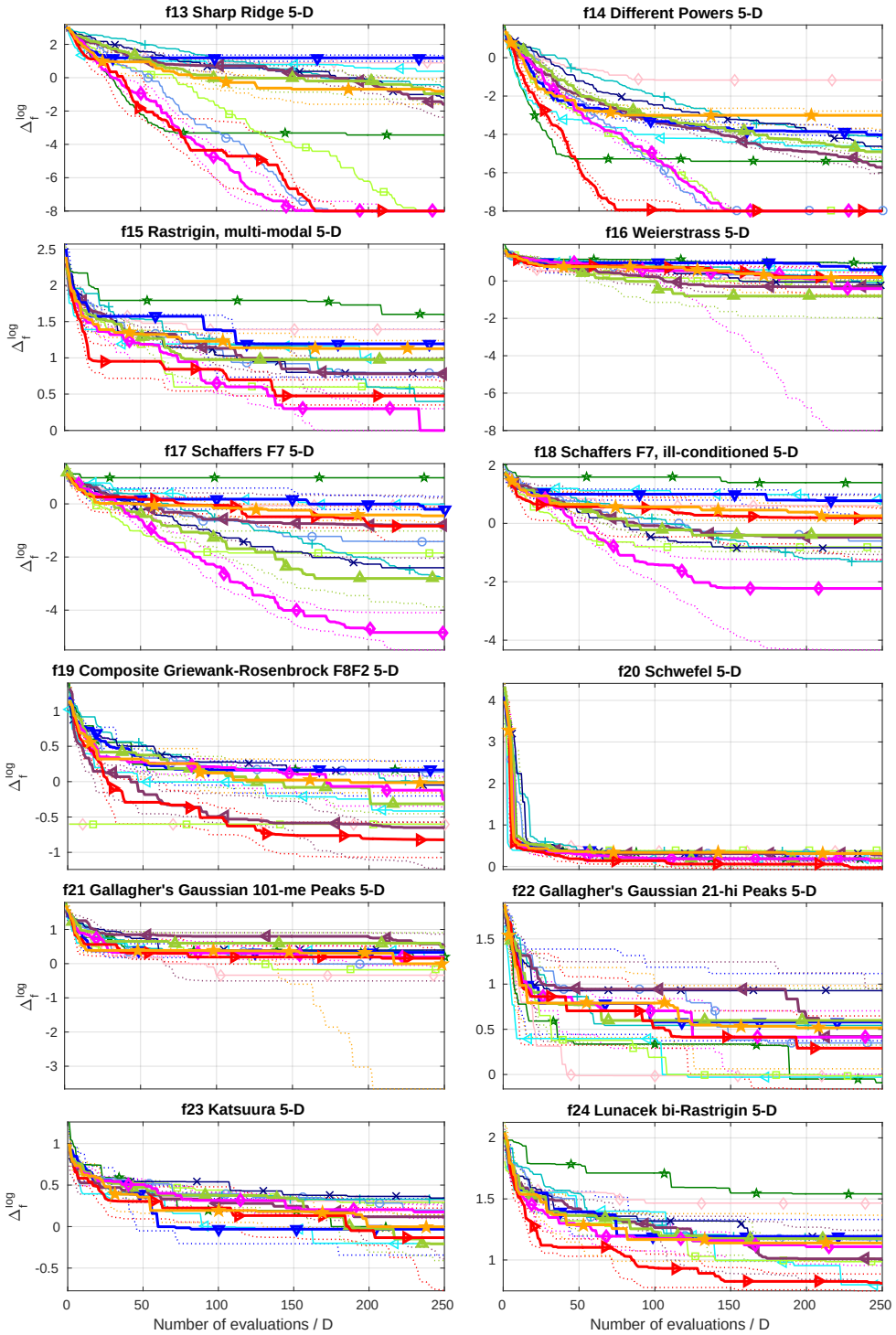


Figure 22: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 24 noiseless COCO benchmarks f_{13-24} in $5D$ for 13 black-box optimizers with the emphasis on the six Gaussian process/CMA-ES algorithms. Medians/quartiles were calculated across 15 independent instances for each algorithm and are shown in the $\log_{10}$ scale.

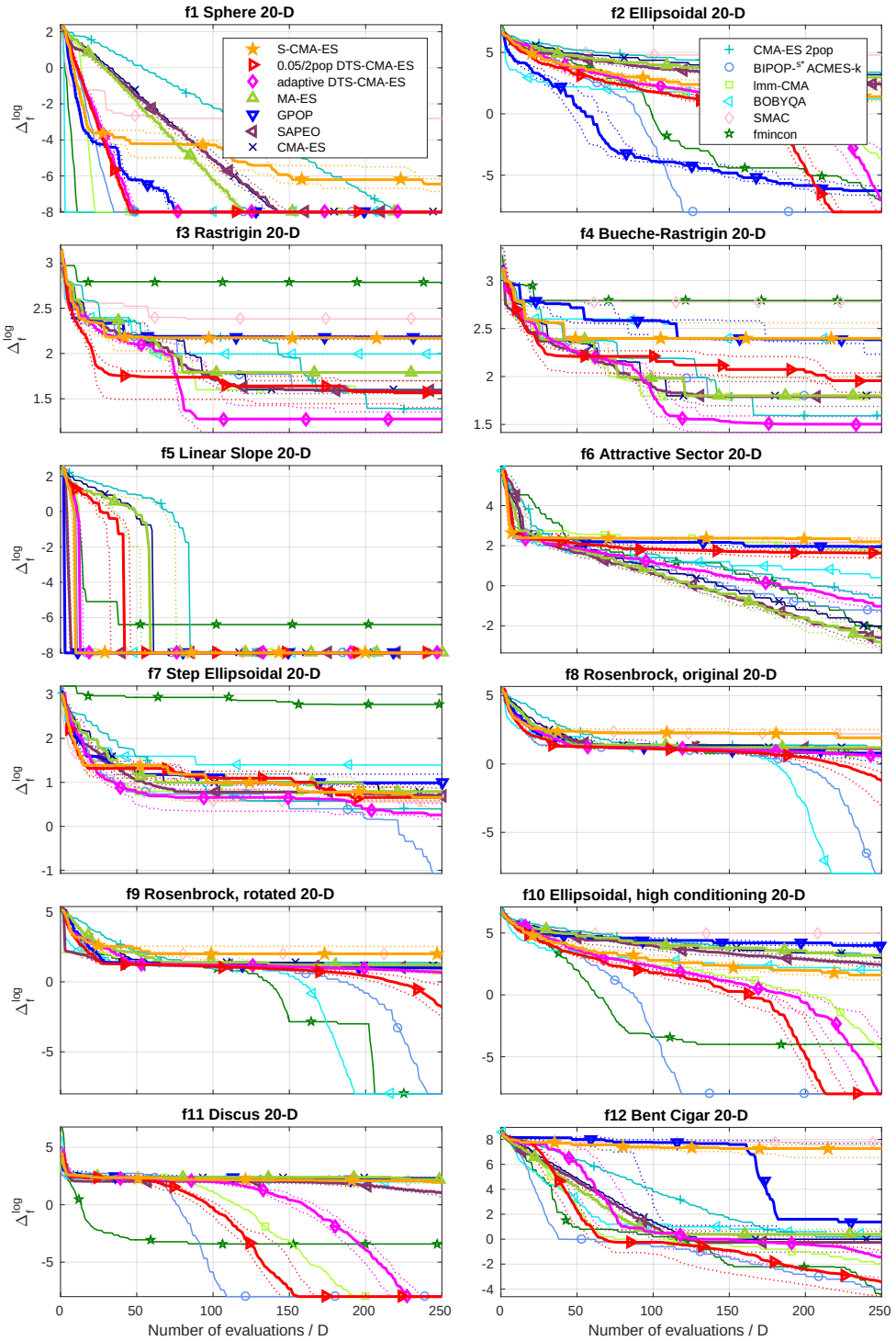


Figure 23: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 12 noiseless COCO benchmarks f_{1-12} in $20D$ for 13 black-box optimizers with the emphasis on the six Gaussian process/CMA-ES algorithms. Medians/quartiles were calculated across 15 independent instances for each algorithm and are shown in the $\log_{10}$ scale.

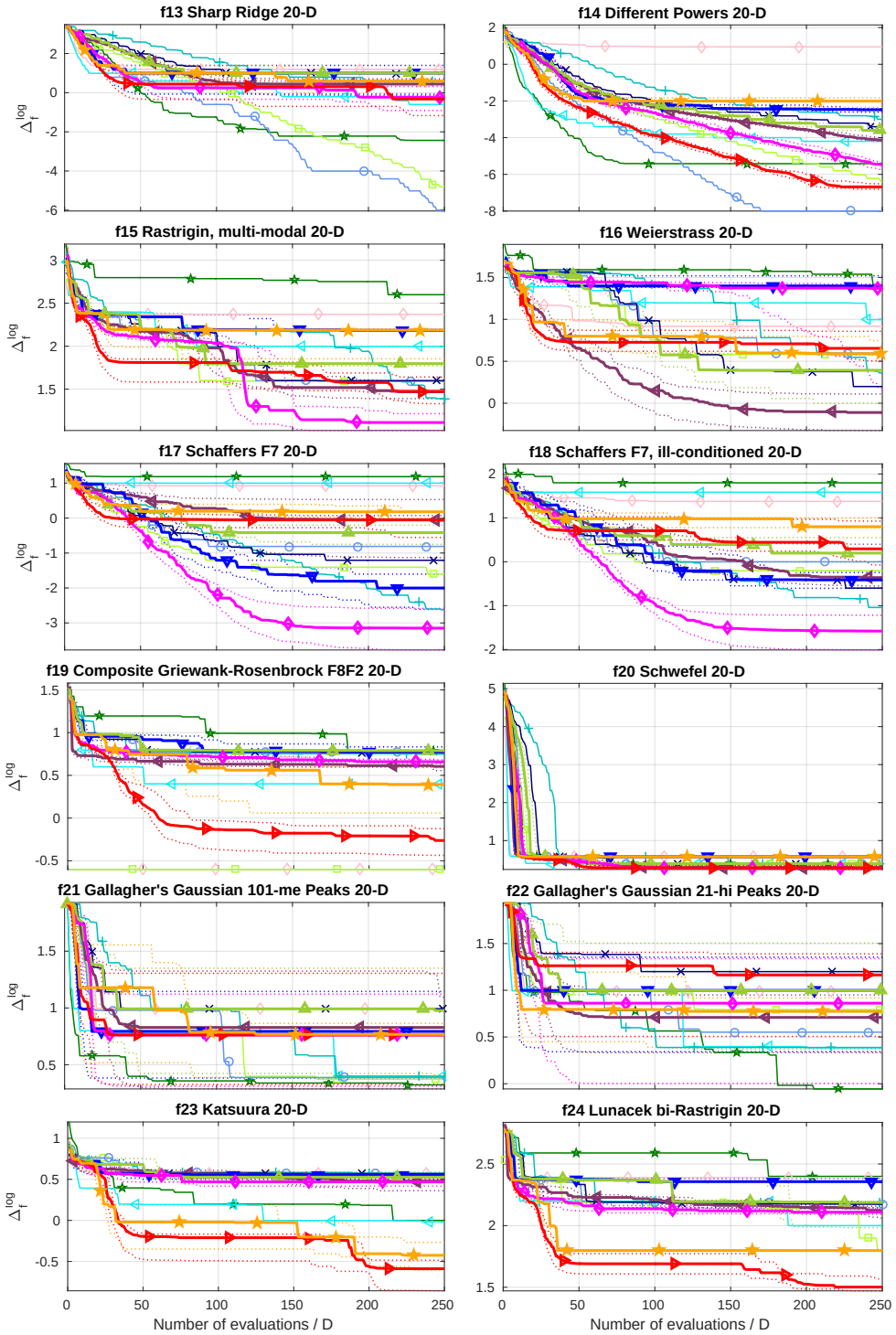


Figure 24: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 12 noiseless COCO benchmarks f_{13-24} in $20D$ for 13 black-box optimizers with the emphasis on the six Gaussian process/CMA-ES algorithms. Medians/quartiles were calculated across 15 independent instances for each algorithm and are shown in the $\log_{10}$ scale.

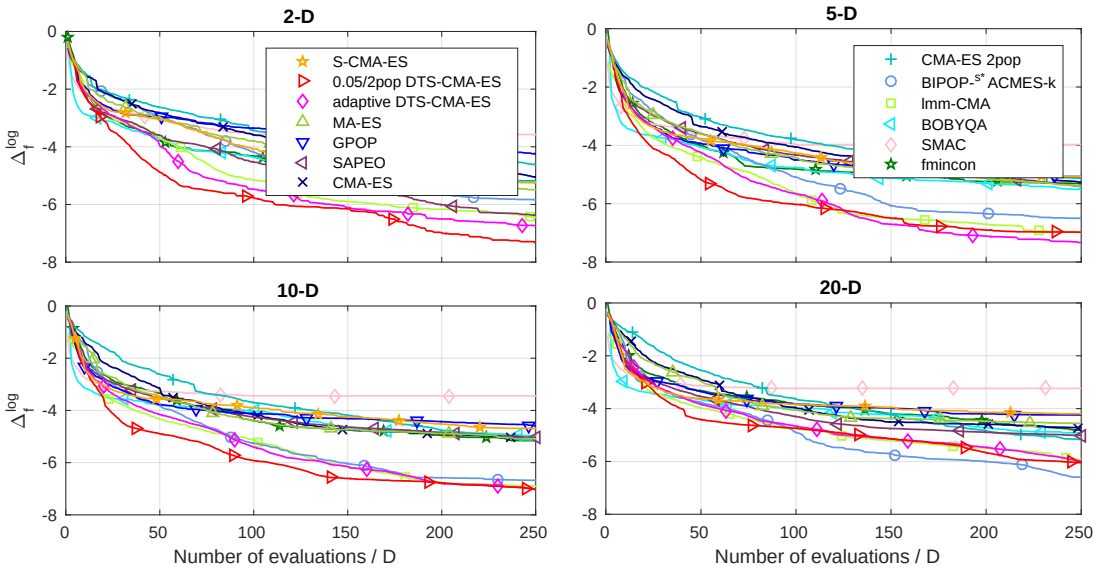


Figure 25: Median distances to the benchmarks' optima averaged over all 24 noiseless COCO functions in dimensions $2, 5, 10, 20$. The $\log_{10}$ of the median distances were linearly scaled to $[-8, 0]$ for each function before calculating the averages.

Table 16: Mean rank of each algorithm over the COCO and the Iman-Davenport statistic for the 4 considered combinations of dimensionalities and evaluation budgets. The Iman-Davenport statistic F_F with the corresponding p-value of the test of equal performance of all algorithms are in the last two rows. See the caption of Table 15 for explanations of the budgets $1/3$ and 1.

Dim	$5D$		$20D$	
FE/D budget	$1/3$	1	$1/3$	1
S-CMA-ES	8.12	8.54	7.67	8.71
0.05 DTS	**3.73**	**3.21**	**4.10**	5.23
adp. DTS	4.77	3.83	4.10	4.85
MA-ES	6.88	7.25	7.38	8.33
GPOP	7.21	8.67	8.29	8.96
SAPEO	7.04	7.25	5.79	5.96
CMA-ES	8.50	8.62	8.50	8.17
CMA-ES 2pop	9.79	8.25	9.21	7.12
^{s*}ACM-ES	8.25	7.17	5.67	**4.62**
lmm-CMA	4.98	5.06	5.56	5.35
BOBYQA	6.21	6.58	7.50	7.12
SMAC	8.06	9.56	9.35	10.10
fmincon	7.46	7.00	7.88	6.46
F_F	5.41	7.42	6.07	5.87
p-val	$3.00 \cdot 10^{-08}$	$7.75 \cdot 10^{-12}$	$1.93 \cdot 10^{-09}$	$4.37 \cdot 10^{-09}$

on average worst. In addition, uncertainty has been shown to help more on the multimodal benchmarks.

Table 17: The CPU time in seconds per one original function evaluation: averaged results over all 24 noiseless COCO functions measured on the computers of the Czech national computational grid MetaCentrum (single-threaded jobs on mostly Intel Xeon-based computers, ranging from 2.0 to 3.5 GHz with 8.9 to 33.9 SPECfp2006 performance per core).

Algorithm	$2D$	$3D$	$5D$	$10D$	$20D$
S-CMA-ES	2.5e−1	1.5e−1	1.7e−1	2.6e−1	1.1
0.05/2pop DTS-CMA-ES	7.3e−1	6.0e−1	7.5e−1	1.7	1.3e+1
adaptive DTS-CMA-ES	5.0e−1	2.7e−1	2.9e−1	9.1e−1	5.6
MA-ES	2.1e−2	4.5e−2	4.8e−2	4.6e−2	7.1e−2
GPOP	9.4e−1	1.2	1.8	3.4	1.2e+1
CMA-ES 2pop	3.2e−3	2.2e−3	1.4e−3	8.4e−4	6.1e−4
BOBYQA	1.8e−3	1.3e−3	9.5e−4	6.2e−4	5.3e−4
fmincon	2.8e−3	1.7e−3	1.1e−3	6.8e−4	4.5e−4

The section also provides an evaluation of different parameter settings. The selection of the k nearest neighbor points as training sets in a specified range for surrogates performed best, a selection method rather rarely encountered in literature. Other parameter settings, used in the experiments due to their good performance, are rather expectable (e. g., a large radius for the selection of training set, the Matérn covariance function, etc.).

The self-adaptive version of the DTS-CMA-ES uses a similar adaptation mechanism of the number of points for the evaluation with the original fitness as the lmm-CMA. This adaptive version also performs similar to the lmm-CMA on many noiseless COCO benchmark functions, which suggests a hypothesis that the self-adaptation decides the algorithm performance to a great extent. Less importantly, the self-adaptive DTS-CMA-ES is the best out of 13 compared algorithms on at least two benchmarks.

The last part of this section empirically compares the available implementations of the algorithms combining the Gaussian processes and the CMA-ES with the CMA-ES itself and five other state-of-the-art black box optimizers. The fastest convergence rate in terms of the original fitness evaluations out of the six compared GP/CMA-ES algorithms was reached by the DTS-CMA-ES with the fixed parameter $\alpha = 0.05$. This DTS-CMA-ES variant represents an algorithm of choice for multimodal functions with weak global structure and is very eligible for unimodal landscapes, too, especially in lower dimensions. The self-adaptive version, on the other hand, excels on the globally decreasing multimodal functions where it outperforms other compared algorithms.

Competitive results on a relatively broad spectrum of the noiseless COCO benchmarks were also provided by the two non-GP surrogate-assisted CMA-ES algorithms. Since the ^{s*}ACM-ES uses a ranking surrogate model, it is valuable especially for functions without continuous derivatives in the optima. It also provides impressive results in $20D$ where it represents the best choice for many unimodal functions while still providing average results on the multimodal ones. It is worth mentioning that the versions without and with the self-adaptation were placed the 1st and 2nd respectively in the two single-objective tracks of the *Black Box Optimization Competition* at the conference GECCO 2017[7].

7 The detailed results can be found on the competition's webpage https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/

Algorithm 11 Ordinal GP model training

Input: $(\mathbf{x}_i, y_i)_{i=1}^N$ — training points,
r — the number of bins for clustering,
$\boldsymbol{\theta}^0$ — initial values of latent GP hyperparameters $\boldsymbol{\theta}$,
α^0, $\{\beta_j^0\}_{j=1}^{r-1}$ — initial values of PLSOR hyperparameters $\alpha, \{\beta_j\}_{j=1}^{r-1}$

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

2: $(\alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})^* \leftarrow \underset{\alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta}}{\arg\max} \log \hat{\mathcal{L}}(\{y_i^{\text{ord}}\}_{i=1}^N | \{\mathbf{x}_i\}_{i=1}^N, \alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta})$ (see Eq. (40))

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

To conclude, the surrogate-model-based CMA-ES algorithms constitute state-of-the-art optimizers for black-box optimization problems. In particular, the Gaussian processes have been shown to be a very competitive (if not the best) surrogate models, especially for multimodal moderately-dimensional objective functions.

4.5 Comparison of Ordinal and Metric Gaussian Process Regression

In this section, we present an approach of using a Gaussian process as an ordinal surrogate model for the CMA-ES from (Pitra et al., 2017b). Although the continuity of Gaussian processes suggests the already presented metric regression, the CMA-ES, due to its invariance with respect to order preserving transformations, might perform better when used with ordinal regression models, similarly to the s*ACM-ES by Loshchilov et al. (2013b) which uses the ordinal Support vector regression. Since a few approaches to the ordinal Gaussian process regression have appeared recently (see Section 2.3.1), we decided to compare PLSOR method also described in Section 2.3.1 with the metric GP regression model, particularly the performance of the models when used as surrogates for the CMA-ES. Because this is (up to our knowledge) the first time the ordinal GP regression is used for surrogate modelling, we have investigated also the suitability of several different settings of the employed ordinal regression method to this end.

The next section sketches some details of our implementation of ordinal GP regression, and, in Section 4.5.2, which is the core part of this investigation, the results of testing both kinds of GP regression in connection with the CMA-ES on the noiseless part of the COCO platform are reported using the COCO-provided graphs.

4.5.1 *Implementation of Ordinal GP*

The ordinal GP model-building phase, depicted in Algorithm 11, starts with clustering the input data $(\mathbf{x}_i, y_i)_{i=1}^N$ transformed by the generalized model training procedure (see Section 4.1.2) to intervals $I_1, \ldots, I_r$. After that, the hyperparameters are selected to maximize the likelihood (40) via leave-one-out cross-validation (Rasmussen and Williams, 2006, p. 111).

The ordinal GP model prediction procedure is depicted in Algorithm 12. The prediction of the ordinal class q_i of a point $\mathbf{x}_i$ is calculated as the expectation of the ordinal class values of $\mathbf{x}_i$ with respect to the probability distribution defined for $\mathbf{x} = \mathbf{x}_i$ according to (39). The output of the GP model is the ordered set of CMA-ES generated population $\{\mathbf{x}_{i:\lambda}\}_{i=1}^{\lambda}$, where

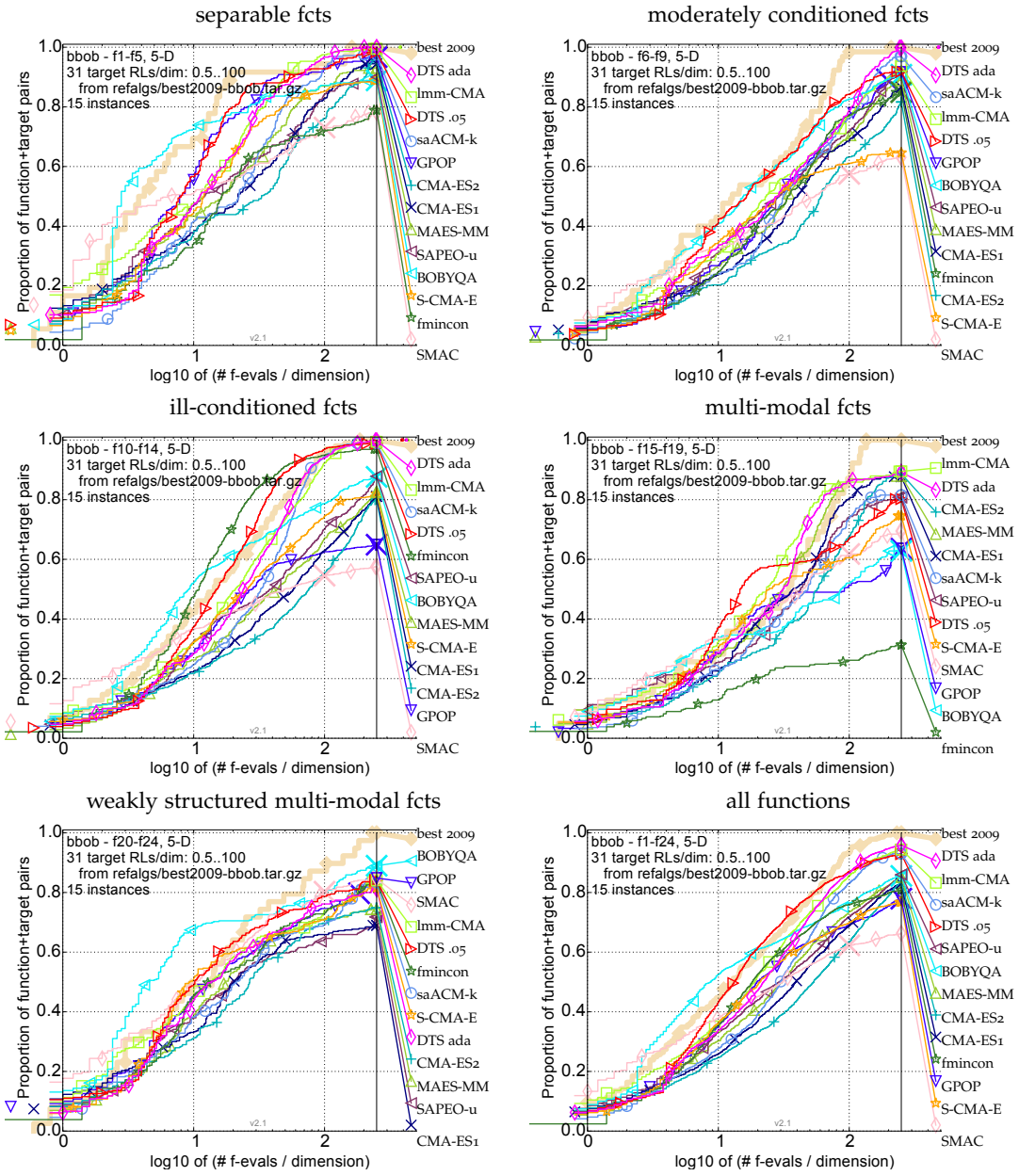


Figure 26: Bootstrapped empirical cumulative distribution of the number of objective FE/D for all functions and subgroups in 5D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times D$, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 100\}$. The "best 2009" curve corresponds to the best *expected runtime* (ERT) observed during BBOB 2009 for each selected target. The ERT depends on a given target function value, $\mathbb{b}_t = \mathbb{b}_{\text{opt}} + \Delta\mathbb{b}$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach $\mathbb{b}_t$, summed over all trials and divided by the number of trials that actually reached $\mathbb{b}_t$ (Hansen et al., 2012).

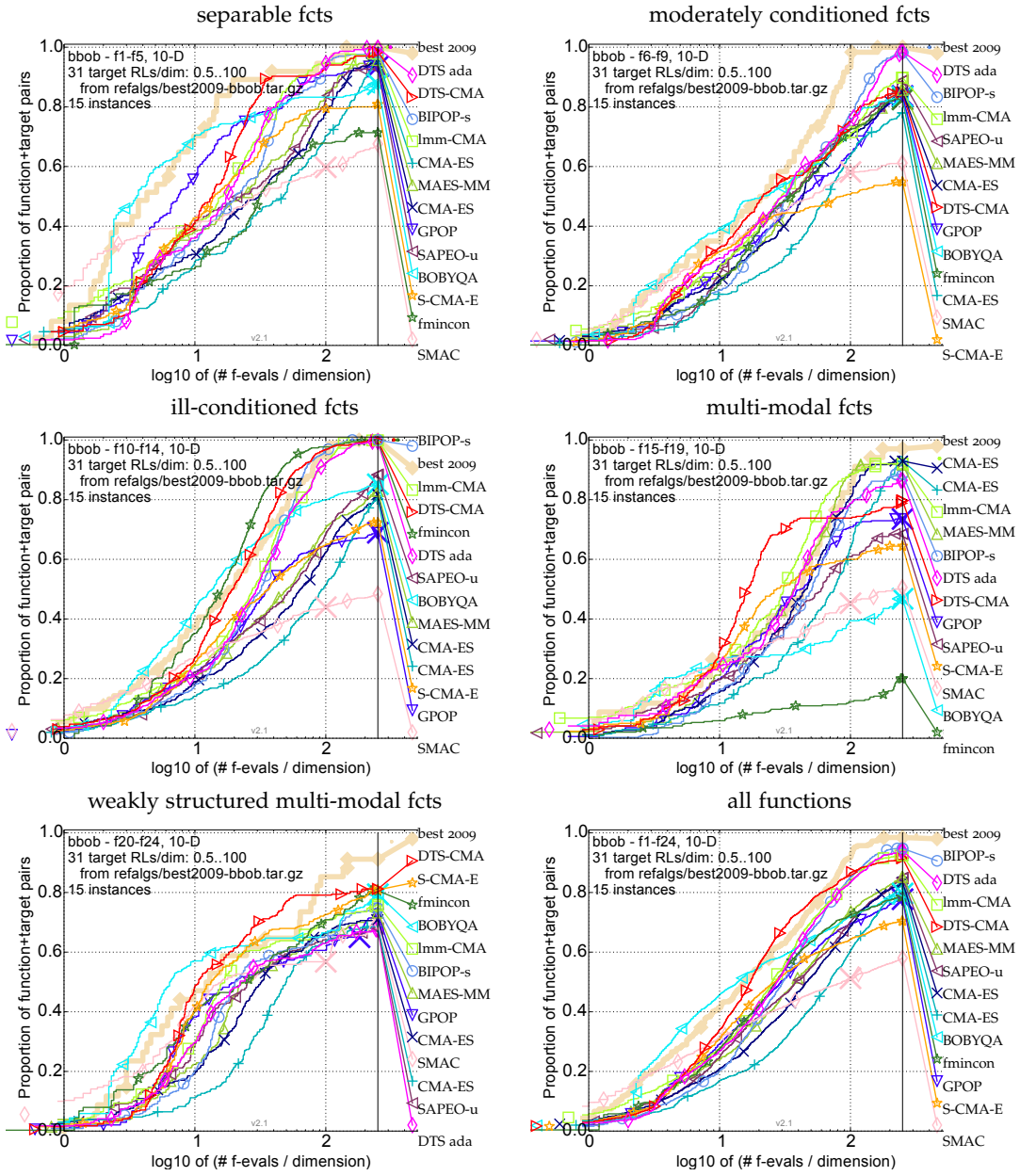


Figure 27: Bootstrapped empirical cumulative distribution of the number of objective FE/D for all functions and subgroups in 10D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times D$, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 100\}$. The "best 2009" curve corresponds to the best *expected runtime* (ERT) observed during BBOB 2009 for each selected target. The ERT depends on a given target function value, $\mathfrak{bb}_t = \mathfrak{bb}_{opt} + \Delta\mathfrak{bb}$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach $\mathfrak{bb}_t$, summed over all trials and divided by the number of trials that actually reached $\mathfrak{bb}_t$ (Hansen et al., 2012).

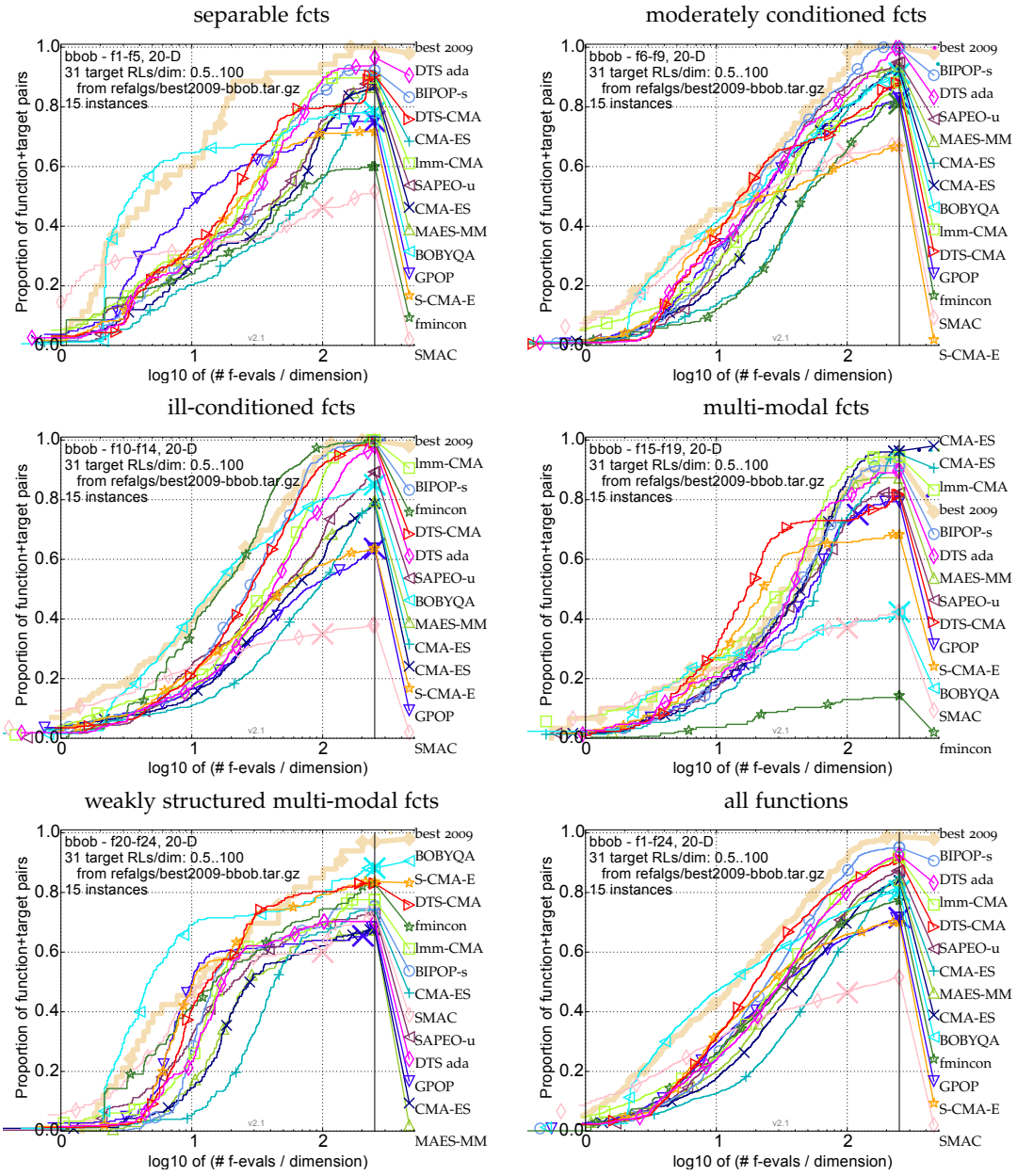


Figure 28: Bootstrapped empirical cumulative distribution of the number of objective FE/D for all functions and subgroups in 20D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times D$, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 100\}$. The "best 2009" curve corresponds to the best *expected runtime* (ERT) observed during BBOB 2009 for each selected target. The ERT depends on a given target function value, $\mathbb{b}_t = \mathbb{b}_{\text{opt}} + \Delta\mathbb{b}$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach $\mathbb{b}_t$, summed over all trials and divided by the number of trials that actually reached $\mathbb{b}_t$ (Hansen et al., 2012).

Algorithm 12 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

1: $p_i^k \leftarrow P(\mathfrak{bb}(\mathbf{x}_i) \in I_k | \mathbf{x}_i, \alpha, \{\beta_j\}_{j=1}^{r-1}, \boldsymbol{\theta}), \quad \forall k = 1, \ldots, r, \quad \forall i = 1, \ldots, \lambda$ (see Eq. (39))

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

3: $\{\mathbf{x}_{i:\lambda}\}_{i=1}^{\lambda} \leftarrow$ order $\{\mathbf{x}_i\}_{i=1}^{\lambda}$ according to $q_{1:\lambda} \leq q_{2:\lambda} \leq \cdots \leq q_{\lambda:\lambda}$

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

the index $i{:}\lambda$ denotes the index of the i-th point ranked, according to the expected order w.r.t. the probability distribution (39).

4.5.2 *Experiments on the COCO platform*

Validation of Our PLSOR Implementation

In order to validate at least experimentally that our PLSOR implementation conforms with the results of already published framework, our code was benchmarked on a collection of 9 datasets from the UCI machine learning repository, similarly to the previous approaches to ordinal regression with Gaussian processes (Chu and Ghahramani, 2005; Srijith et al., 2012a).

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,

$$\mathrm{err}_{\mathrm{ZOE}} = \frac{1}{t} \sum_{i=1}^{t} \mathbb{I}\left(y_i \neq \hat{y}_i\right) , \tag{87}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Table 18 compares ZOE means and standard deviations on the 5-categories versions of the benchmark datasets with the results reported in (Srijith et al., 2012a).

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 differences between both implementations for 5- and 10-categories versions are not significant ($p = 0.16, 0.34$ respectively). The validation was performed with the squared exponential covariance κ_{SE} with initial parameter values and other implementation choices matching those given in (Srijith et al., 2012a) when possible.

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 $f_i, i \in \{1, 2, \ldots, 24\}$ and for dimensions $D \in \{2, 5, 10\}$ from 10 snapshots of the DTS-CMA-ES *archive* $\mathcal{A}$ (i. e., the set of so-far originally-evaluated points). These snapshots were taken equidistantly throughout CMA-ES

Table 18: 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 (Srijith et al., 2012a) (middle column) and values for our implementation (right column). The lower mean values are highlighted in bold.

Data	ZOE (original (Srijith et al., 2012a))	ZOE (our implementation)
Diabetes	$\mathbf{0.48} \pm 0.11$	0.57 ± 0.11
Pyrimidine	0.39 ± 0.09	$\mathbf{0.36} \pm 0.07$
Triazines	0.54 ± 0.03	$\mathbf{0.54} \pm 0.03$
Wisconsin	$\mathbf{0.66} \pm 0.03$	0.68 ± 0.05
Machine	$\mathbf{0.18} \pm 0.03$	0.19 ± 0.04
AutoMPG	$\mathbf{0.26} \pm 0.02$	0.26 ± 0.02
Boston	$\mathbf{0.25} \pm 0.03$	0.25 ± 0.03
Stocks	0.11 ± 0.02	$\mathbf{0.11} \pm 0.02$
Abalone	0.22 ± 0.03	$\mathbf{0.22} \pm 0.04$

generations and each testing dataset was sampled using a simple combination of the CMA-ES state variables $(\mathbf{m}, \sigma, \mathbf{C})$, 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:

1. covariance function – κ_{SE} (used in (Srijith et al., 2012a)) and $\kappa_{\text{Mat}}^{5/2}$;
2. type of obtaining ordinal from continuous $\mathfrak{y}$-value – quantile clustering, agglomerative hierarchical clustering, or direct ordering of $\mathfrak{y}$-values (no clustering);
3. the number of clusters in 2. – μ, λ or 2λ.

The results for $D = 5$ are presented in Table 19. As can be seen, the PLSOR models in general produce a higher ZOE than the metric GP. An exception is the datasets for the ATTRACTIVE SECTOR function f_6 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 w.r.t. the smoothness of the corresponding function.

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 noiseless benchmarking in connection with the DTS-CMA-ES algorithm:

- ◇ *Ord-N-DTS*, N denotes no clustering at all,
- ◇ *Ord*-Q-*DTS*, Q stands for quantile-based clustering, and
- ◇ *Ord*-H-*DTS*, H is 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 fitness function re-evaluation. The DTS-CMA-ES with both ordinal and metric models was used in the version very similar to the version published in (Pitra

Table 19: Means and standard deviations of ZOE of offline testing on 10 selected generations for 24 noiseless COCO functions and 5D, DTS − GP: a metric GP regression model, κ_{SE} and $\kappa_{\text{Mat}}^{5/2}$: 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.

5D	DTS − GP	$\kappa_{\text{Mat}}^{5/2}$	$\kappa_{\text{Mat}}^{5/2}$, H, μ	$\kappa_{\text{Mat}}^{5/2}$, H, λ	$\kappa_{\text{Mat}}^{5/2}$, H, 2λ	$\kappa_{\text{Mat}}^{5/2}$, Q, μ	$\kappa_{\text{Mat}}^{5/2}$, Q, λ	$\kappa_{\text{Mat}}^{5/2}$, Q, 2λ	κ_{SE}	κ_{SE}, H, μ	κ_{SE}, H, λ	κ_{SE}, H, 2λ	κ_{SE}, Q, μ	κ_{SE}, Q, λ	κ_{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	$\mathbf{0.76 \pm 0.18}$	0.82 ± 0.21
f_2	$\mathbf{0.73 \pm 0.18}$	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	$\mathbf{0.84 \pm 0.10}$	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	$\mathbf{0.75 \pm 0.38}$	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	$\mathbf{0.00 \pm 0.00}$	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	$\mathbf{0.55 \pm 0.32}$	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	$\mathbf{0.81 \pm 0.06}$	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	$\mathbf{0.74 \pm 0.23}$	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	$\mathbf{0.70 \pm 0.28}$	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	$\mathbf{0.78 \pm 0.19}$	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}	$\mathbf{0.81 \pm 0.12}$	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}	$\mathbf{0.67 \pm 0.32}$	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}	$\mathbf{0.69 \pm 0.28}$	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	$\mathbf{0.82 \pm 0.12}$	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	$\mathbf{0.78 \pm 0.31}$	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	$\mathbf{0.84 \pm 0.16}$	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	$\mathbf{0.85 \pm 0.12}$	0.93 ± 0.07
f_{18}	$\mathbf{0.82 \pm 0.18}$	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	$\mathbf{0.62 \pm 0.26}$	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}	$\mathbf{0.63 \pm 0.33}$	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}	$\mathbf{0.70 \pm 0.38}$	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}	$\mathbf{0.70 \pm 0.36}$	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	$\mathbf{0.68 \pm 0.38}$	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}	$\mathbf{0.77 \pm 0.22}$	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
	$\mathbf{0.74 \pm 0.29}$	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

et al., 2016). The remaining parameters were left the same as in this original DTS-CMA-ES settings, too: the GP prediction variance as the uncertainty criterion, the population size $\lambda = 8 + \lfloor 6 \ln D \rfloor$, and the ratio of originally-evaluated points $\alpha = 0.05$.

The original CMA-ES was employed in its IPOP-CMA-ES version (Matlab code v. 3.61) using settings identical to Bajer et al. (2015) described in Section 4.2.1.

The results in Figure 29 show the effect of using the PLSOR models instead of the metric GP in the DTS-CMA-ES optimizer on all the 24 noiseless COCO benchmarks (Hansen et al., 2009b, 2016) (see ECDF graphs in more detail in Section 2.6.1). The experiments were conducted with the maximum budget of 100 FE/D in dimensions $D = 2, 3, 5, 10$. Experiments in higher dimensions were not performed due to immense computational requirements. More detailed results can be found on an authors' webpage[8].

The effect of different clustering methods seems not to be important (see Figure 29). 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}), whereas the original DTS-CMA-ES is dominant on the remaining tested functions and in higher dimensions.

We have tested the statistical significance of performance differences in $5D$ using the Iman and Davenport's improvement of the Friedman test (Demšar, 2006). The test is conducted separately for two function evaluation budgets, where the maximum number of FE/D is only 100, compared to the previous chapter. The algorithms are ranked on each COCO function with respect to Δf^{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 "1" ($p < 10^{-3}$), as well as at a lower budget "1/3" ($p < 10^{-3}$) (see caption of Table 15 for the definition of these budgets; 250 FE/D is replaced with 100 FE/D here).

We test pairwise differences in performance of the algorithms using the post-hoc Friedman test (García and Herrera, 2008) with a control of the family-wise error: Bergmann-Hommel procedure corrects the significance level per each logically consistent family of hypotheses. The results of these multiple comparisons are reported in Table 20. 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.

CPU Timing

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

4.5.3 *Conclusion*

In this section, we have compared the ordinal GP regression model using PLSOR implementation with the metric GP regression model used in the DTS-CMA-ES. The comparison of our implementation of the PLSOR method reproduced the published results on the UCI datasets, but the use of the PLSOR models as surrogates for the CMA-ES was not shown as straightforward on the COCO benchmark: the performance of the PLSOR models is considerably

8 https://jdgregorian.github.io/surrogate-cmaes-pages/supp/gecco2017bbob/

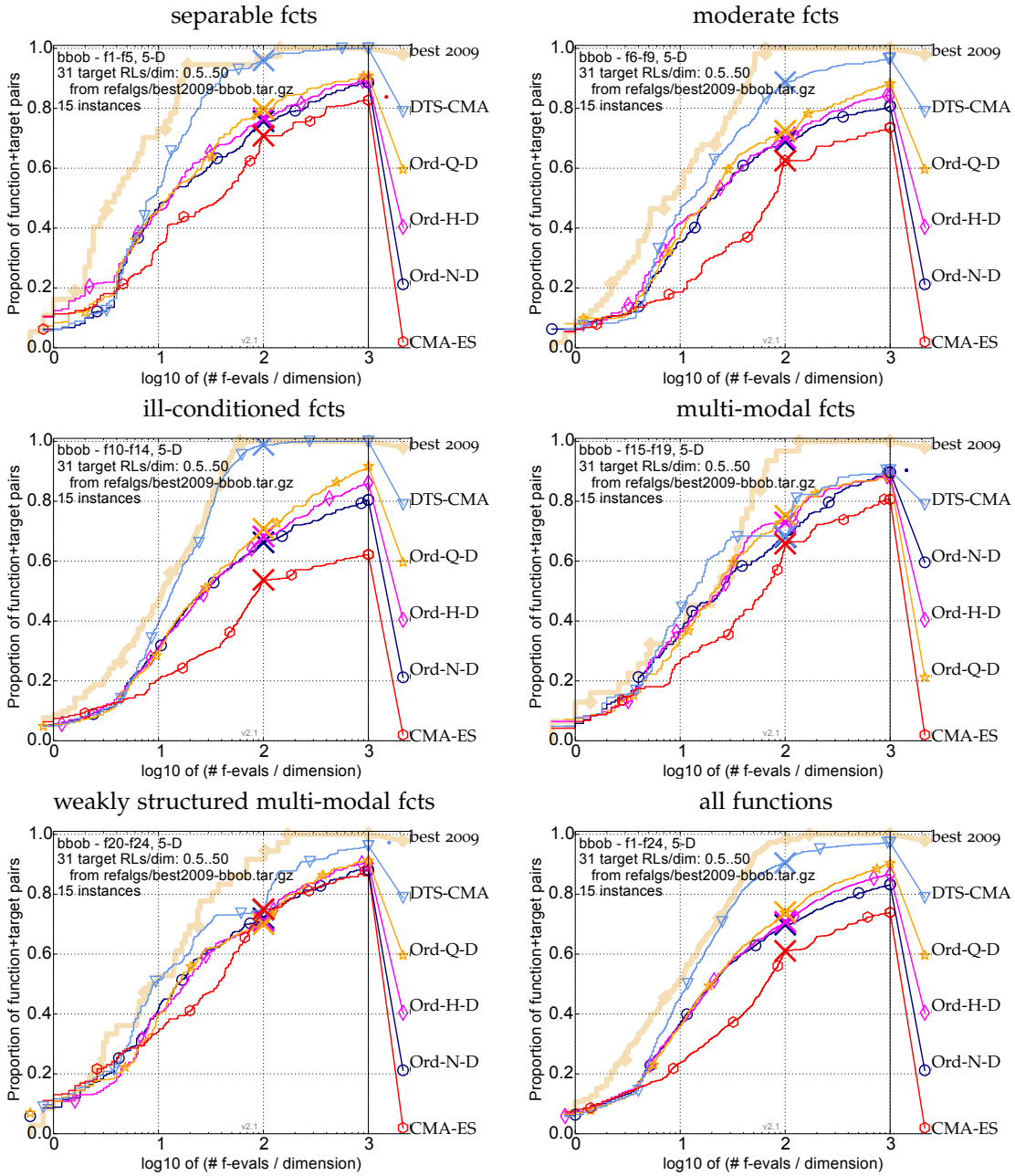


Figure 29: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for all functions and subgroups in $5D$. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times D$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" curve corresponds to the best ERT observed during BBOB 2009 for each selected target.

Table 20: A pairwise comparison of the algorithms in 5D on the COCO noiseless functions 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 achieving a significantly lower value of the objective function than the column algorithm (on medians over 15 instances taken from all 24 functions) according to the Friedman post-hoc test with the Bergmann-Hommel correction at the family-wise significance level $\alpha = 0.05$. The double asterisk marks additional 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	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	—	—

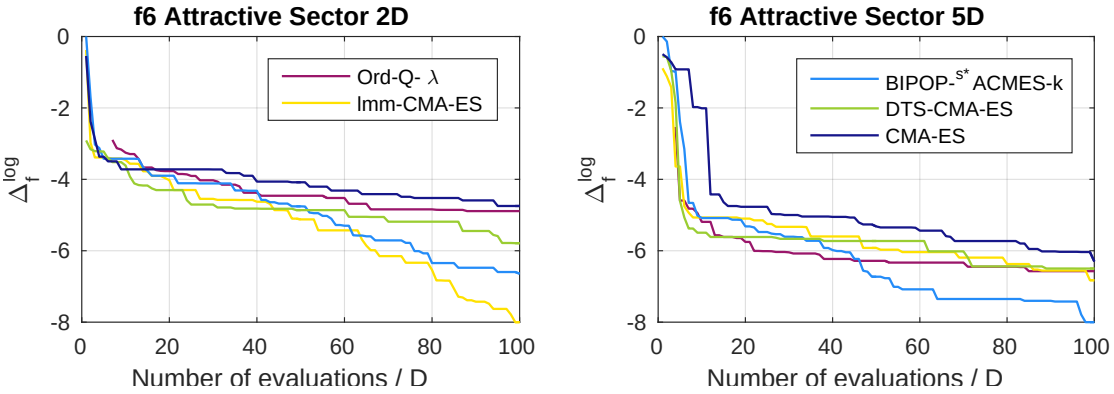


Figure 30: Comparison of optimization algorithms on the Attractive sector function f_6 in 2D and 5D.

lower than the standard GP models with few exceptions, especially on the Attractive sector function f_6 (see Figure 30).

4.6 Automated selection of covariance function for Gaussian process surrogate models

A principal choice in specifying a Gaussian process model is the choice of the covariance function, which largely embodies the prior assumptions about the modeled function. Several methods for learning the form of covariance function have been proposed:

- Learning a composite expression of kernel functions for support vector machines by genetic programming was explored in (Gagné et al., 2006).

- ◇ Hierarchical kernel learning (Bach, 2009) and Additive Gaussian processes (Duvenaud et al., 2011) are algorithms for determining kernels composed of lower-dimensional kernels.
- ◇ The goal of Automatic Statistician project (Lloyd et al., 2014) is automatic statistical analysis of given data with output in natural language.
- ◇ The algorithm of structure discovery in GP models (Duvenaud et al., 2013) is a greedy search in the space of composite covariance functions generated by operators of addition and multiplication recursively applied to basis covariance functions.

Up to our knowledge, structure discovery in GP surrogate models for continuous black-box optimization has not yet been investigated. As a first step towards this goal, in (Repický et al., 2018a,b), we performed selection of the best GP model from a model population that we tried to design large enough to capture structure of typical continuous black-box function but still small enough for model selection to be computationaly feasible. The main hypothesis behind this research is that a GP with a composite form of its covariance function may result in a more accurate approximation of the objective function and, consequently, better performance of the model-assisted optimization algorithm.

Hierarchical Model

When the GP covariance function family is given, model selection for GP regression is usually performed by maximum marginal likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \ln p(\mathbf{y}_N \mid \mathbf{X}_N, \boldsymbol{\theta})$, where $p(\mathbf{y}_N \mid \mathbf{X}_N, \boldsymbol{\theta})$ is a marginal likelihood from (15), which is a non-convex optimization problem.

From a Bayesian perspective, especially if the number of hyperparameters is large or if number of observations N is small, it might be more appropriate to do inference with the marginal posterior distribution of hyperparameters

$$p(\boldsymbol{\theta} \mid \mathbf{X}_N, \mathbf{y}_N) = \frac{p(\mathbf{y}_N \mid \mathbf{X}_N, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y}_N \mid \mathbf{X}_N)}, \tag{88}$$

where $p(\mathbf{y}_N \mid \mathbf{X}_N, \boldsymbol{\theta})$ is the marginal likelihood (15), now playing the role of the likelihood, and $p(\boldsymbol{\theta})$ is a hyper-prior. Simulations from $p(\boldsymbol{\theta} \mid \mathbf{X}_N, \mathbf{y}_N)$ can be obtained by Bayesian computation methods, such as Markov chain Monte Carlo.

4.6.1 *Model Selection*

If the probability of the true value of the fitness function conditioned on the GP prior is low, the performance of the model will be poor. For example, a GP with a neural network covariance κ_{NN} (see Equation (31)) fits data from a jump function better compared to a GP with a squared exponential (more on in Section 2.3.1). Searching over GP models with different covariances thus can be viewed as an automated construction of suitable priors. We select the model from a finite set according to a criterion of predictive performance, since this approach can easily be embedded into a combinatorial search algorithm, such as in (Duvenaud et al., 2013).

Performance Criteria

We would like to select the surrogate model based on an estimation of out-of-sample predictive accuracy.

An attractive estimate of the out-of-sample predictive accuracy is cross-validation based on some partitioning of the data set into multiple data sets called folds. An efficient algorithm for hyperparameter optimization with leave-one-out cross-validation has been developed by Sundararajan and Keerthi (2001). However, choosing among multiple GP models by cross-validation in each generation of the evolutionary optimization can be considered prohibitive from the computational perspective.

In the remainder of this subsection, we follow the exposition of model comparison from Bayesian perspective given in (Gelman et al., 2014). We denote by q the true distribution from which data $\mathbf{y}_N$ are sampled and we suppress conditioning on $\mathbf{X}_N$ for simplicity.

A general measure of fit of a probabilistic model $\mathbf{y}_N$ to data is the log likelihood or log predictive density $\ln p(\mathbf{y}_N \mid \boldsymbol{\theta}) = \ln \prod_{i=1}^{N} p(y_i \mid \boldsymbol{\theta})$. The quantity $-2 \ln p(y \mid \boldsymbol{\theta})$ is called deviance.

The *Akaike information criterion* (AIC) (Akaike, 1973) and the related *Bayes information criterion* (BIC) (Schwarz, 1978) are based on the expected log predictive density conditioned on a maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$,

$$\mathrm{elpd}_{\hat{\boldsymbol{\theta}}} = \mathbb{E}_q(\ln p(\tilde{\mathbf{y}}_N \mid \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}))\,, \tag{89}$$

where the expectation is taken over all possible data sets $\tilde{\mathbf{y}}_N$. Since expectation (89) cannot be computed exactly, it is estimated from sample $\mathbf{y}_N$. The AIC and Bayes information criterion (BIC) compensate for bias towards overfitting by substracting a correction term, the number of parameters $n_{\boldsymbol{\theta}}$ and $\frac{1}{2} n_{\boldsymbol{\theta}} \ln N$, respectively.

More precisely, the AIC is written as:

$$\mathrm{AIC} = -2 \ln p(\mathbf{y}_N \mid \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}) + 2 n_{\boldsymbol{\theta}}\,. \tag{90}$$

The closely related BIC is based on approximation of marginal likelihood and uses a bias correction that is dependent on the sample size:

$$\mathrm{BIC} = -2 \ln p(\mathbf{y}_N \mid \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}) + n_{\boldsymbol{\theta}} \ln N\,. \tag{91}$$

Compared to AIC, the BIC tends to favor models with lower number of parameters for $n_{\boldsymbol{\theta}} \geq 8$.

For hierarchical Bayesian models, such as (88), it is not always entirely clear, what the parameters of the model are, since the likelihood can factorize in different ways. The *deviance information criterion* (DIC) (Spiegelhalter et al., 2002) is still based on deviance, conditioned on a Bayes estimate $\hat{\boldsymbol{\theta}}_{\mathrm{Bayes}}$, but the effective number of parameters p_{DIC} depends on data. We define the DIC for the marginal likelihood (15), focusing on hyperparameters $\boldsymbol{\theta}$, although it could be defined for the likelihood $p(\mathbf{y}_N \mid \mathfrak{w}, \boldsymbol{\theta})$, focusing on both $\mathfrak{w}$ and $\boldsymbol{\theta}$.

The original definition of p_{DIC} is the difference between posterior mean of the deviance and the deviance conditioned on posterior mean of parameters under focus (Spiegelhalter et al., 2002). We use the following definition of the effective number of parameters (see (Gelman et al., 2014)):

$$p_{\mathrm{DIC}} = 2\mathrm{var}_{\mathrm{post}}(\ln p(\mathbf{y}_N \mid \boldsymbol{\theta}))\,, \tag{92}$$

which can be estimated by the sample variance of a posterior sample. Using the effective number of parameters, the DIC is

$$\mathrm{DIC} = -2 \ln p(\mathbf{y}_N \mid \hat{\boldsymbol{\theta}}_{\mathrm{Bayes}}) + 2 p_{\mathrm{DIC}}\,. \tag{93}$$

A probabilistic model is called regular if its parameters are identifiable and its Fisher information matrix is positive definite for all parameter values. The model is called singular

otherwise. The information criteria defined above assume regularity. Watanabe (2007) proved that many machine learning models are singular and proposed the the *Widely applicable information criterion* (WAIC) (Watanabe, 2010) that works also for singular models. The WAIC is based on estimation of the expected log *pointwise* predictive density

$$\text{elppd} = \sum_{i=1}^{N} \mathbb{E}_q(\ln p_{\text{post}}(\tilde{y}_i)) = \sum_{i=1}^{N} \mathbb{E}_q \left(\ln \int p(\tilde{y}_i \,|\, \mathbf{y}_N, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \mathbf{y}_N) \, \mathrm{d}\boldsymbol{\theta} \right) . \tag{94}$$

The estimation of elppd from the sample is biased, so again, an effective number of parameters must be added as a correction. We use the following definition of the WAIC (see (Gelman et al., 2014)):

$$\text{WAIC} = -\sum_{i=1}^{N} \ln p_{\text{post}}(y_i) + \sum_{i=1}^{N} \text{var}_{\text{post}}(\ln p(y_i \,|\, \boldsymbol{\theta})) , \tag{95}$$

that is the negative log pointwise predictive density corrected for bias by pointwise posterior variance of log predictive density.

The pointwise predictive density $p_{\text{post}}(y_i \,|\, \mathbf{y}_N, \boldsymbol{\theta})$ for the GP model (15) is computed by integrating Gaussian likelihood over the marginal posterior GP at i^{th} training point:

$$\begin{aligned} p(y_i \,|\, \mathbf{y}_N, \boldsymbol{\theta}) &= \int p(y_i \,|\, \mathbf{y}_N, \mathfrak{bb}_i, \boldsymbol{\theta}) p(\mathfrak{bb}_i \,|\, \mathbf{y}_N, \boldsymbol{\theta}) \, \mathrm{d}\mathfrak{bb}_i \\ &= \varphi(y_i \,|\, \hat{\mathfrak{bb}}_i, \sigma_n^2 + \text{var}(\mathfrak{bb}_i)) , \end{aligned}$$

where φ denotes the Gaussian density and $\hat{\mathfrak{bb}}_i, \text{var}(\mathfrak{bb}_i)$ represent $\hat{\mathbf{y}}^*, \hat{\Sigma}^*$ in (22) respectively.

4.6.2 *Experimental Evaluation*

In this section, we describe preliminary experimental evaluation of DTS-CMA-ES that uses a GP model with an automated selection of covariance function. Since GPs are a nonparametric model, we opt for the WAIC, which require a sample from distribution (88). We use Metropolis-Hastings Markov chain Monte Carlo with an adaptive proposal distribution (Haario et al., 2006)[9].

Algorithm 9 is updated in the following way[10]:

1. In steps 4 and 9, all GPs from a user defined portfolio are trained.
2. The predictive accuracy of all models is evaluated using the specified information criterion.
3. The model with the lowest WAIC is used for prediction (steps 5 and 10).

Experimental Setup

The proposed algorithm implemented in MATLAB was evaluated on the noiseless part of the COCO/BBOB framework (Hansen et al., 2009b, 2012) and compared with the original DTS-CMA-ES using a GP surrogate according to Algorithm 9 and the original CMA-ES itself. We run the algorithm on 5 instances $(1 - 5)$ as opposed to 15 recommended instances for the reason of increased computational demands of the modified algorithm. For the same reason, we have performed evaluation only in $10D$ with the budget of $250\,\text{FE}/D$.

9 Using MATLAB implementation available at http://helios.fmi.fi/~lainema/dram/
10 The source codes are available at https://github.com/repjak/surrogate-cmaes/tree/modelsel

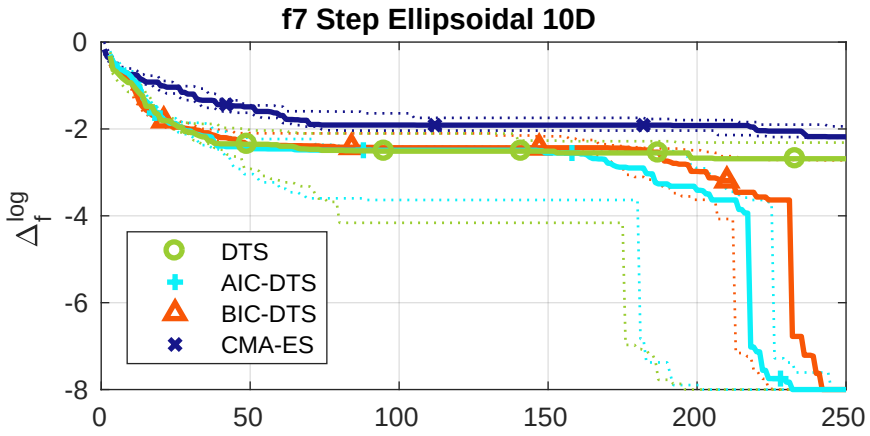


Figure 31: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima against the number of function evaluations on the Step Ellipsoid COCO function f_7 in $10D$ for the CMA-ES, the DTS-CMA-ES, and the DTS-CMA-ES with the adaptive covariance selection according to AIC and BIC: AIC-DTS, BIC-DTS. The medians and quartiles were calculated from 15 independent runs on different function instances. Distances to optima are shown in the $\log_{10}$ scale (see Section 2.6.2 for details).

CMA-ES The CMA-ES results in BBOB format were downloaded from the BBOB 2010 workshop archive[11].

DTS-CMA-ES If not stated otherwise, the DTS-CMA-ES was employed in its overall best adaptive settings from (Bajer et al., 2019) (see Section 4.4).

DTS-CMA-ES WITH IC We have employed the following information criteria: AIC, BIC, and WAIC. The DIC (93) values were also computed during the run together with WAIC for information, but not taken into any algorithm decision due to high computational demands. The selection was performed among the following covariance functions: κ_{LIN}, κ_{Q}, κ_{SE}, κ_{NN}, κ_{ADD}, and two composite covariances $\kappa_{\text{SE+Q}}$ and $\kappa_{\text{SE+NN}}$. The hyper-priors are chosen as follows: log-normal with mean $\ln(0.01)$ and variance 2 for σ_n^2; and log-$t_{\nu=4}$ with mean 0 for all other hyperpameters.

Results

The results are shown in Figures 31 and 32 and Table 21. Figures 31 and 32 gives the scaled best-achieved logarithms $\Delta_f^{\log}$ of median distances to the functions optimum for the respective number of FE/D (see Section 2.6.2). Medians and the 1^{st} and 3^{rd} quartiles are calculated from 5 independent instances in case of the algorithm with covariance selection according to the WAIC and from 15 independent instances otherwise.

AIC and BIC optimization results did not show an improvement on the DTS-CMA-ES with no clear difference between them. Nevertheless, a promising result has been obtained on the Step ellipsoid function f_7, characterized by plateaus lying on a quadratic structure, especially

11 http://coco.gforge.inria.fr/data-archive/bbob/2010/

Table 21: Average model ranks on the noiseless part of the COCO testbed in 10D for DIC and WAIC predictive performance criteria selecting covariance function for the DTS-CMA-ES. The lowest values for the considered combination of COCO function and performance criterion are in bold.

Criterion	WAIC							DIC						
Model	κ_{SE}	κ_{NN}	κ_{LIN}	κ_{Q}	κ_{ADD}	$\kappa_{\text{SE+NN}}$	$\kappa_{\text{SE+Q}}$	κ_{SE}	κ_{NN}	κ_{LIN}	κ_{Q}	κ_{ADD}	$\kappa_{\text{SE+NN}}$	$\kappa_{\text{SE+Q}}$
f_1	3.64	4.01	6.94	4.17	**2.73**	3.25	3.27	5.57	4.68	6.65	3.11	**1.55**	4.07	2.37
f_2	3.07	3.05	6.96	5.41	4.35	**2.48**	2.67	5.38	4.85	6.78	3.99	**1.45**	3.71	1.84
f_3	2.94	3.20	6.75	5.99	4.14	**2.45**	2.53	4.37	4.63	6.74	5.40	**1.40**	3.11	2.36
f_4	3.00	3.20	6.87	5.73	4.11	**2.52**	2.57	4.49	4.67	6.76	5.21	**1.30**	3.27	2.30
f_5	3.69	3.60	6.78	4.41	**2.69**	3.28	3.56	6.75	4.16	4.71	3.25	3.48	3.00	**2.66**
f_6	3.03	3.09	6.99	5.95	3.99	**2.46**	2.50	4.18	4.62	6.92	5.92	1.86	2.80	**1.69**
f_7	3.15	3.09	6.92	5.86	3.91	**2.49**	2.58	4.35	4.87	6.90	5.37	**1.54**	2.99	1.98
f_8	2.83	3.12	6.97	5.76	4.15	**2.50**	2.66	4.78	4.57	6.83	5.21	**1.32**	3.27	2.02
f_9	2.86	3.14	6.97	5.69	4.14	2.63	**2.57**	4.86	4.57	6.79	5.05	**1.32**	3.37	2.04
f_{10}	3.00	3.16	6.96	5.43	4.31	**2.52**	2.62	5.39	4.82	6.76	3.95	**1.53**	3.79	1.75
f_{11}	3.01	3.09	6.97	5.53	4.24	**2.57**	2.60	5.21	5.17	6.80	3.66	2.14	3.87	**1.16**
f_{12}	3.16	3.28	6.93	4.37	4.96	**2.64**	2.66	5.46	4.65	6.66	3.57	**1.46**	3.94	2.27
f_{13}	2.93	2.98	6.98	5.83	4.42	**2.37**	2.50	4.87	4.47	6.88	5.23	**1.24**	3.06	2.25
f_{14}	3.29	2.96	7.00	5.90	3.63	**2.59**	2.64	4.26	4.84	6.99	5.64	**1.43**	3.04	1.81
f_{15}	2.86	3.28	6.73	5.95	4.13	2.58	**2.47**	4.25	4.69	6.80	5.58	**1.48**	2.92	2.28
f_{16}	2.53	3.59	6.46	6.50	4.16	2.41	**2.36**	3.69	4.69	6.78	6.10	**1.80**	2.44	2.51
f_{17}	3.25	3.05	6.86	6.09	3.52	**2.58**	2.65	4.03	4.69	6.90	5.95	**1.36**	2.80	2.26
f_{18}	3.17	3.05	6.88	6.10	3.65	**2.55**	2.60	3.98	4.70	6.88	6.00	**1.42**	2.74	2.28
f_{19}	2.60	3.52	6.42	6.54	4.20	**2.32**	2.39	3.69	4.67	6.75	6.13	**1.45**	2.52	2.78
f_{20}	2.98	3.37	6.94	5.55	4.01	2.62	**2.51**	4.46	4.73	6.81	5.22	**1.30**	3.26	2.21
f_{21}	3.14	3.21	6.87	5.07	4.61	2.59	**2.51**	4.91	4.65	6.76	4.56	**1.34**	3.47	2.31
f_{22}	3.11	3.22	6.88	5.00	4.57	**2.58**	2.63	5.06	4.60	6.73	4.31	**1.41**	3.51	2.37
f_{23}	2.47	3.77	6.35	6.58	4.28	2.34	**2.21**	3.50	4.76	6.68	6.15	**1.74**	2.47	2.70
f_{24}	2.73	3.53	6.44	6.51	4.06	2.38	**2.34**	3.76	4.68	6.77	6.11	**1.42**	2.53	2.73

in multi-dimensional variants (see Figure 31). The most frequently selected kernel for this function under both AIC and BIC has been the sum of the squared exponential and the quadratic kernel, which provides an intuitive interpretation of the function.

We observe that in most cases, the WAIC-based algorithm mostly barely outperforms the pure CMA-ES, which suggests the chosen model is generally weak and the adaptivity mechanism basically turns off using the surrogate model. The functions where the WAIC variant outperforms the DTS-CMA-ES (f_{21} and f_{22}) are multi-modal and the interquartile range is large.

In order to obtain the notion of DIC results, we compare the rank of each model under WAIC and DIC. Table 21 summarizes the average ranks over all model selections performed on each benchmark function. We observe that the DIC often prefers the additive model, while the WAIC is more balanced in this respect. Surprisingly the linear covariance κ_{LIN} has been very rarely selected even on the Linear slope function f_5 under both information criteria. A similar observation holds for the quadratic kernel κ_{Q} and the quadratic functions Sphere f_1 and Ellipsoid f_2.

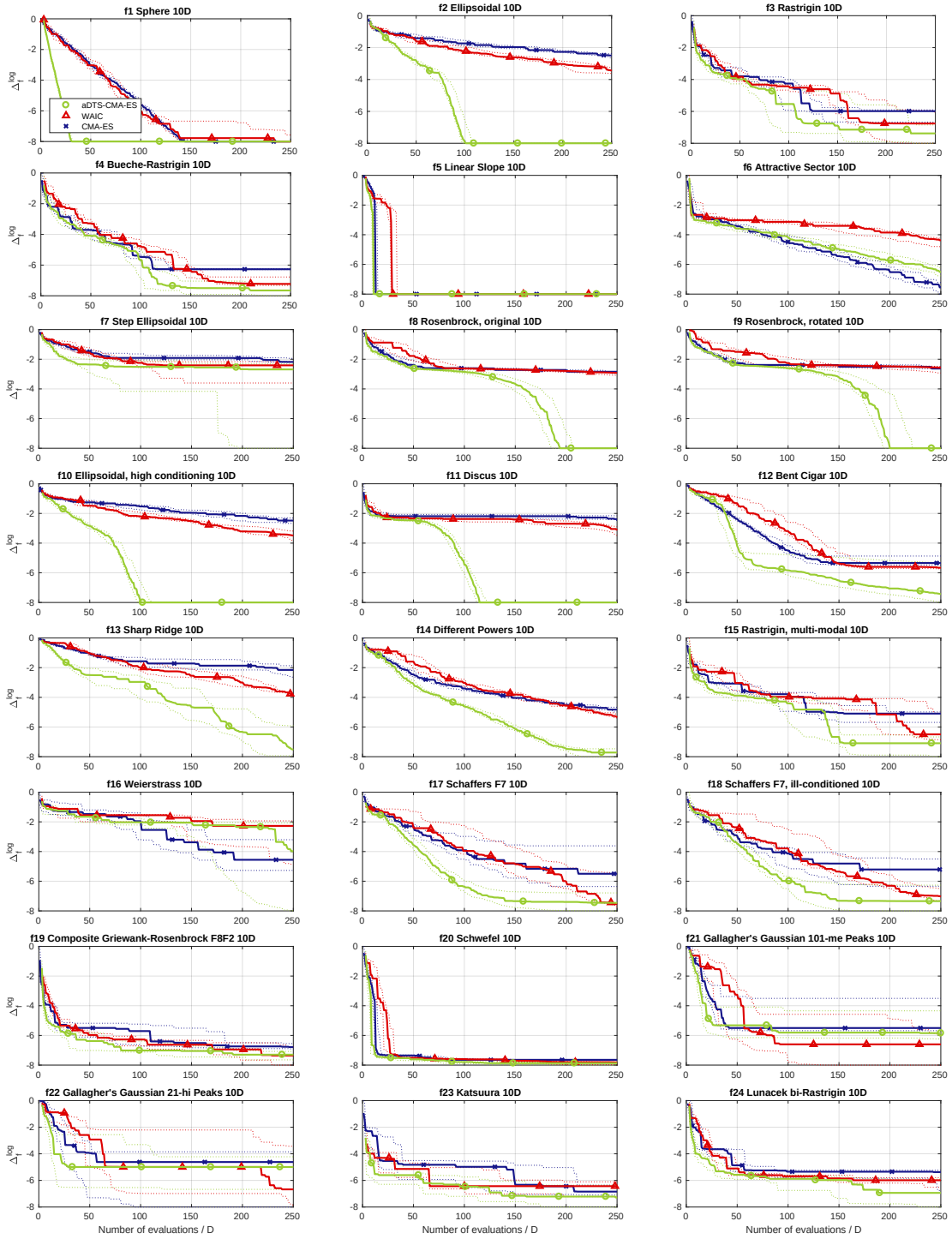


Figure 32: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 24 COCO/BBOB benchmarks in $10D$ for algorithms DTS-CMA-ES (green, denoted aDTS-CMA-ES), DTS-CMA-ES with WAIC-based model selection (red) and CMA-ES (blue). The medians and quartiles for WAIC variant were calculated from 5 independent instances. In all other cases, 15 indepenent instances were used. Distances to optima are shown in the $\log_{10}$ scale (see Section 2.6.2).

4.6.3 *Conclusion*

In this section, we presented an algorithm for selecting a GP model using Bayesian model comparison techniques. Preliminary experiments for the model selection plugged into the DTS-CMA-ES algorithm were conducted on the COCO/BBOB testbed. Due to the small number of experiments performed so far, it is difficult to draw any serious conclusions. The first obtained results may indicate improper convergence of the Markov chain Monte Carlo sampler or that more sophisticated covariance functions may be needed.

One direction of future research, beside analyzing and repairing aforementioned deficiencies, is an extension of the proposed algorithm into a combinatorial search over kernels in flavor of (Duvenaud et al., 2013; Gagné et al., 2006), which is challenging due to computational costs related to the need of repeated surrogate model retraining.

One possible direction of research is a *co-evolution* of an ensemble of covariance functions alongside the population of candidate solutions to the black-box objective function. Other related research area is applying surrogate modeling to high-dimensional problems using algorithms for variable selection via multiple kernel learning (Bach, 2009; Duvenaud et al., 2011).

4.7 BOOSTED REGRESSION FOREST FOR THE DTS-CMA-ES

In (Pitra et al., 2018a), we have studied the DTS-CMA-ES in connection with the boosted regression forest (see Boosting in Section 2.3.1), another regression model capable to estimate the distribution. Results of testing regression forest and Gaussian processes, the former in 20 different settings, as a surrogate models in the DTS-CMA-ES on the set of noiseless benchmarks are reported in the following text.

4.7.1 *Evaluation of Boosted Regression Forest for the DTS-CMA-ES*

In this section, we compare the performances of the DTS-CMA-ES using the RFs as a surrogate model in several different settings to the original DTS-CMA-ES version, the original CMA-ES, and the lmm-CMA on the noiseless part of the COCO platform (Hansen et al., 2009b, 2012).

Experimental setup

The considered algorithms were compared on 24 noiseless single-objective continuous benchmark functions from the COCO testbed (Hansen et al., 2009b, 2012) in dimensions $D = 2, 3, 5$, and 10 on 15 different instances per function. Each algorithm had a budget of $250D$ function evaluations to reach the target distance $\Delta f_T = 10^{-8}$ from the function optimum. The parameter settings of the tested algorithms are summarized in the following paragraphs.

The original CMA-ES was employed in its IPOP-CMA-ES version (Matlab code v. 3.61) using settings identical to (Bajer et al., 2015) described in Section 4.2.1.

The lmm-CMA was utilized in its improved version published by Auger et al. (2013). The results have been downloaded from the COCO results data archive[12] in its GECCO 2013 settings.

12 http://coco.gforge.inria.fr/data-archive/2013/lmm-CMA-ES_auger_noiseless.tgz

Table 22: Experimental settings of RF: n_{orig} – number of originally-evaluated points , n_{tree} – number of trees in RF, N_t, n_D – number of tree points and dimensions. Split methods and n_{orig} are selected using full-factorial design, n_{tree}, N_t, and n_D are sampled.

parameter	values
n_{orig}	$\{\lceil 0.05\lambda \rceil, \lceil 0.1\lambda \rceil, \lceil 0.2\lambda \rceil, \lceil 0.4\lambda \rceil\}$
split	{CART, SECRET, OC1, SUPPORT, PAIR}
n_{tree}	$\{64, 128, 256, 512, 1024\}$
N_t	$\lceil \{0.25, 0.5, 0.75, 1\} \cdot N \rceil$
n_D	$\lceil \{0.25, 0.5, 0.75, 1\} \cdot D \rceil$

The original DTS-CMA-ES was tested using the overall best settings from (Bajer et al., 2019) (see Section 4.4.3): the PoI as the uncertainty criterion, the population size $\lambda = 8 + \lfloor 6 \ln D \rfloor$, and the number of originally-evaluated points $n_{\text{orig}} = \lceil 0.05\lambda \rceil$.

Considering decision tree settings, the five splitting methods from the following algoritms were employed: CART (Breiman, 1984), SECRET (Dobra and Gehrke, 2002), OC1 (Murthy et al., 1994), SUPPORT (Chaudhuri et al., 1994), and decision tree splitting method based on pairs of points (PAIR) (Hinton and Revow, 1996). Due to the different properties of individual splitting methods, the number of err evaluations was limited to $10D$ per node to restrict the algorithms which test a great number of hyperplanes. For the same reason, the number of tresholds generated by a projection of points to a hyperplane was set to 10 quantile-based values in CART, OC1, and to a median value in PAIR, and the searching an initial axis-aligned hyperplane in OC1 was limited to $\lceil \frac{10D}{3} \rceil$ err evaluations.

The RFs as a surrogate model were tested using the gradient boosting ensemble method. The maximum tree depth was set to 8, in accordance with (Chen and Guestrin, 2016). In addition, the number of trees n_{tree}, the number of points N_t bootstrapped out of N archive points, and the number of randomly subsampled dimensions used for training the individual tree n_D were sampled from the values in Table 22.

The DTS-CMA-ES in combination with RFs was tested with the following settings: the PoI as the uncertainty criterion, the population size $\lambda = 8 + \lfloor 6 \ln D \rfloor$, and the number of originally-evaluated points n_{orig} with 4 different values $\lceil 0.05\lambda \rceil$, $\lceil 0.1\lambda \rceil$, $\lceil 0.2\lambda \rceil$, and $\lceil 0.4\lambda \rceil$. The rest of DTS-CMA-ES parameters have been taken identical to the overall best settings from (Bajer et al., 2019) (see Section 4.4.3).

Results

Result from experiments are presented in Figures 33–36 and also in Table 23 (see Section 2.6.2 for detailed explanation of convergence graphs).

We compare the statistical significance of differences in algorithms' performance on 24 COCO functions in $5D$ for separately two evaluation budgets utilizing the Iman and Davenport's improvement of the Friedman test (Demšar, 2006). Let $\#\text{FE}_\text{T}$ be the smallest number of FEs on which at least one algorithm reached the target distance, i. e., satisfied $\Delta_f^{\text{med}} \leq \Delta f_T$, or $\#\text{FE}_\text{T} = 250D$ if no algorithm reached the target within $250D$ evaluations. The algorithms are ranked on each function with respect to Δ_f^{med} at a given budget of FEs. The null hypothe-

sis of equal performance of all algorithms is rejected for the higher function evaluation budget #FEs = #FE$_\mathrm{T}$ ($p < 10^{-3}$), as well as for the lower budget #FEs = $\frac{\#\mathrm{FE}_\mathrm{T}}{3}$ ($p < 10^{-3}$).

We test pairwise differences in performance utilizing the post-hoc test to the Friedman test (García and Herrera, 2008) with the Bergmann-Hommel correction controlling the family-wise error. The numbers of functions at which one algorithm achieved a higher rank than the other are enlisted in Table 23. The table also contains the pairwise statistical significances.

The graphs in Figures 34 and 35 summarize the performance of five different split algorithms and four n_orig values from twenty different settings respectively. We found that the convergence of DTS-CMA-ES is quite similar regardless the split algorithm with slightly better results of SECRET and SUPPORT – the algorithms utilizing classification methods to find the splitting hyperplane between previously created clusters of training points. The results also show that lower n_orig values provide better performance in the initial phase of the optimization run and higher values are more successful starting from the $100 - 150\,\mathrm{FE}/D$. Due to the presented results, the following comparisons contain the performances of the DTS-CMA-ES with $n_\mathrm{orig} = \lceil 0.4\lambda \rceil$ in combination with RF using SECRET and SUPPORT as split algorithms.

As can be seen in Figures 33 and 36, the performance of RFs is considerably worse than the performance of GPs in combination with the DTS-CMA-ES and better than the performance of the original CMA-ES. RF model provides faster convergence from approximitely $100\,\mathrm{FE}/D$ on the regularly multimodal Rastrigin functions (f_3, f_4, and f_{15}) where the RF apparently does not prevent the original CMA-ES from exploiting the global structure of a function. The performance of RF-DTS-CMA-ES is noticably lower especially on the Ellipsoid (f_1, f_2, f_7, and f_{10}), Rosenbrock (f_8, f_9), and ill-conditioned functions (f_{11-14}), where smooth models are much more convenient for regression. On the other hand, RFs help the CMA-ES to convergence especially on the multimodal functions f_{16-19}, where the performance of RF-DTS-CMA-ES is the best of all compared algorithms.

4.7.2 *Conclusion*

In this section, we have compared the RF model using gradient boosting as the ensemble method with the GP regression model, both used as surrogate models in the DTS-CMA-ES algorithm. Different methods of space splitting in regression trees were investigated.

The split algorithms SECRET and SUPPORT based on the classification of the input points provide slightly better performance as to the CMA-ES convergence than the other algorithms tested. Moreover, the performance of DTS-CMA-ES using RFs differs according to the number of originally-evaluated points: the lower their number, the sooner the algorithm converges, possibly to a local optimum, which makes convergence to the global one more difficult. We found that the RF model usually reduces the number of fitness evaluations required by the CMA-ES, especially on multi-modal functions, where the provided speed-up was the best among all compared algorithms for a number of evaluations higher than approximitely $110\,\mathrm{FE}/D$.

4.8 Interaction between model and its EC in surrogate-assisted CMA-ES

Each surrogate modelling method has two complementary aspects: the employed regression model and its evolution control (Jin et al., 2001, 2002; Büche et al., 2005; Loshchilov et al., 2012; Na et al., 2012; Bajer et al., 2019; Chugh et al., 2019; Pitra et al., 2019b). However, EC is typically tightly interconnected with the way how the regression model is trained and subsequently

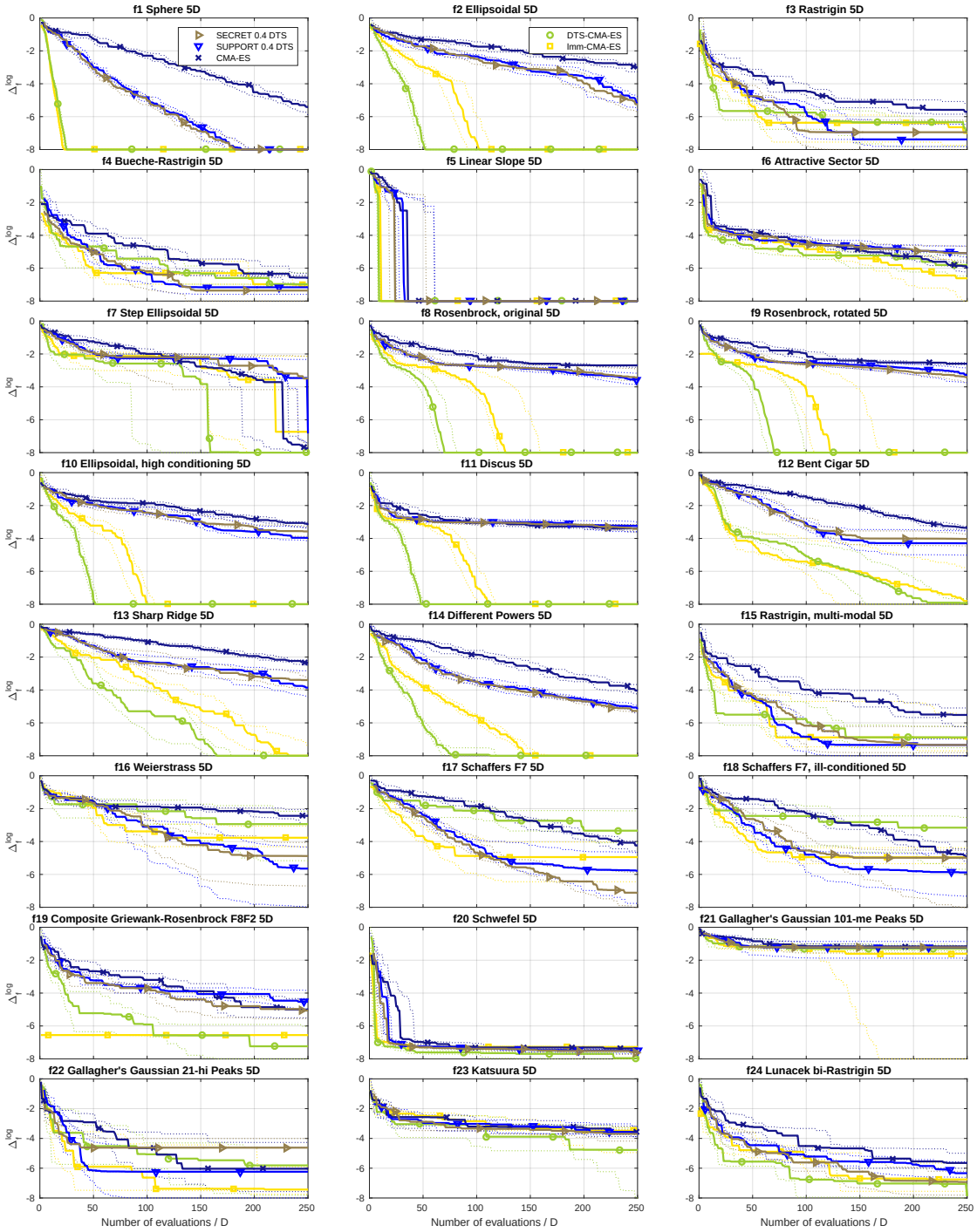


Figure 33: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 24 noiseless COCO benchmarks in $5D$ for algorithms CMA-ES, DTS-CMA-ES, lmm-CMA, and 2 RF settings of DTS-CMA-ES. Medians/quartiles were calculated across 15 independent instances for each algorithm and are shown in the $\log_{10}$ scale.

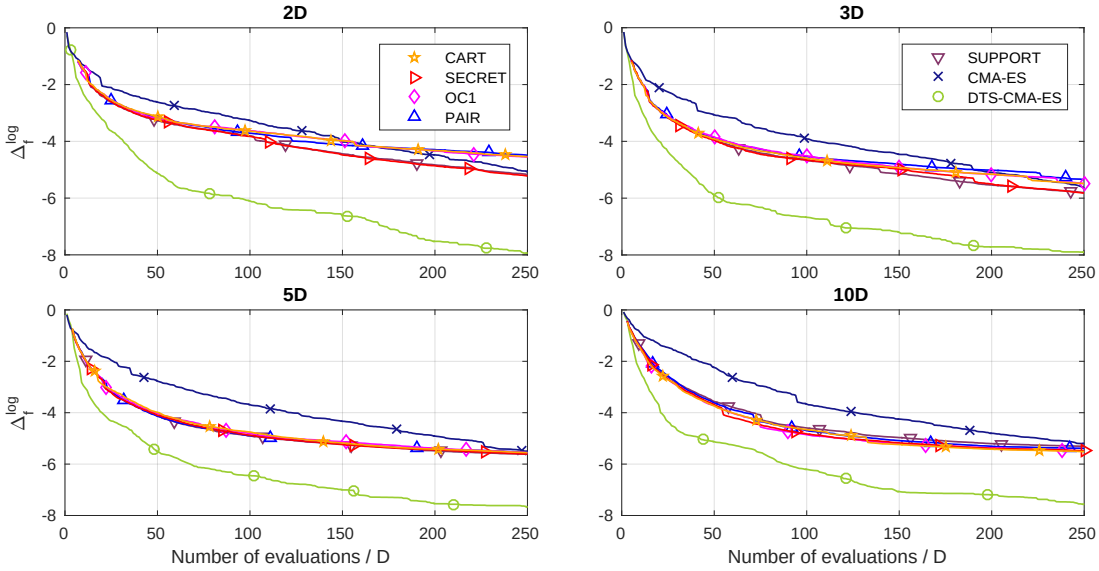


Figure 34: Scaled median distances $\Delta_f^{\log}$ of decision tree split algorithms averaged over all 24 COCO functions in $2D$, $3D$, $5D$, and $10D$ for algorithms CART, SECRET, OC1, PAIR, and SUPPORT in combination with the DTS-CMA-ES and all tested numbers of originally-evaluated points.

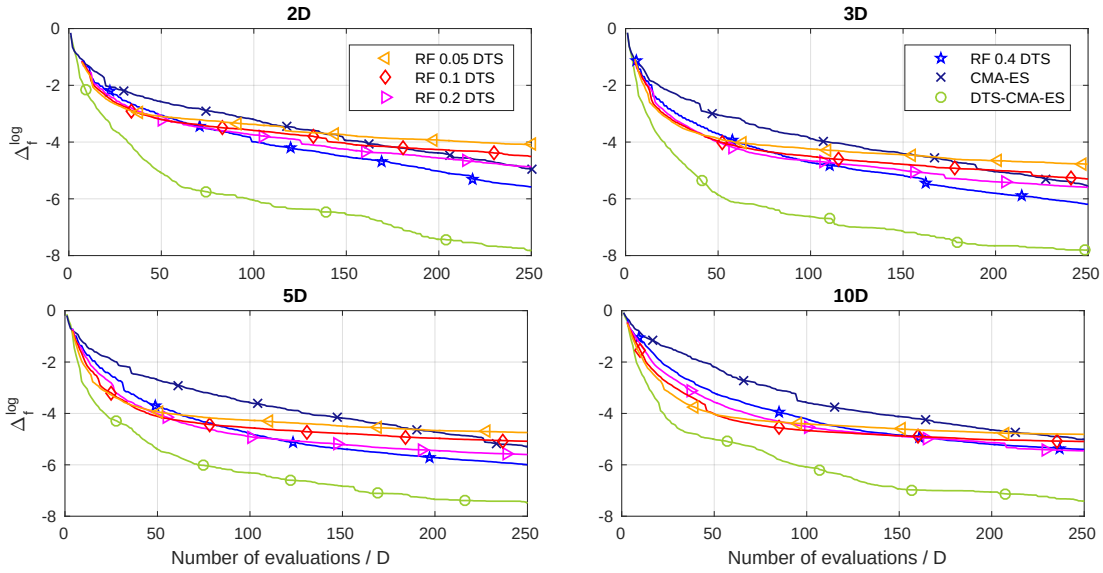


Figure 35: Scaled median distances $\Delta_f^{\log}$ of the DTS-CMA-ES with RFs comparing different numbers of originally-evaluated points averaged over all 24 COCO functions in $2D$, $3D$, $5D$, and $10D$ for values $\lceil 0.05\lambda \rceil$, $\lceil 0.1\lambda \rceil$, $\lceil 0.2\lambda \rceil$, and $\lceil 0.4\lambda \rceil$ summarized accross all tested splitting algorithms.

Table 23: A pairwise comparison of the algorithms in 5D over the COCO 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 Bergmann-Hommel correction at family-wise significance level $\alpha = 0.05$.

5*D*	SECRET 0.4 DTS		SUPPORT 0.4 DTS		CMA-ES		DTS-CMA-ES		lmm-CMA	
$\#FEs/\#FE_T$	1/3	1	1/3	1	1/3	1	1/3	1	1/3	1
SECRET 0.4 DTS	—	—	11.5	11	23*	21*	8	6	5	8
SUPPORT 0.4 DTS	12.5	13	—	—	24*	21*	7	7	7	8
CMA-ES	1	3	0	3	—	—	3	4	1	3
DTS-CMA-ES	16	18	17	17	21*	20*	—	—	14	14
lmm-CMA	19	16	17	16	23*	21*	10	10	—	—

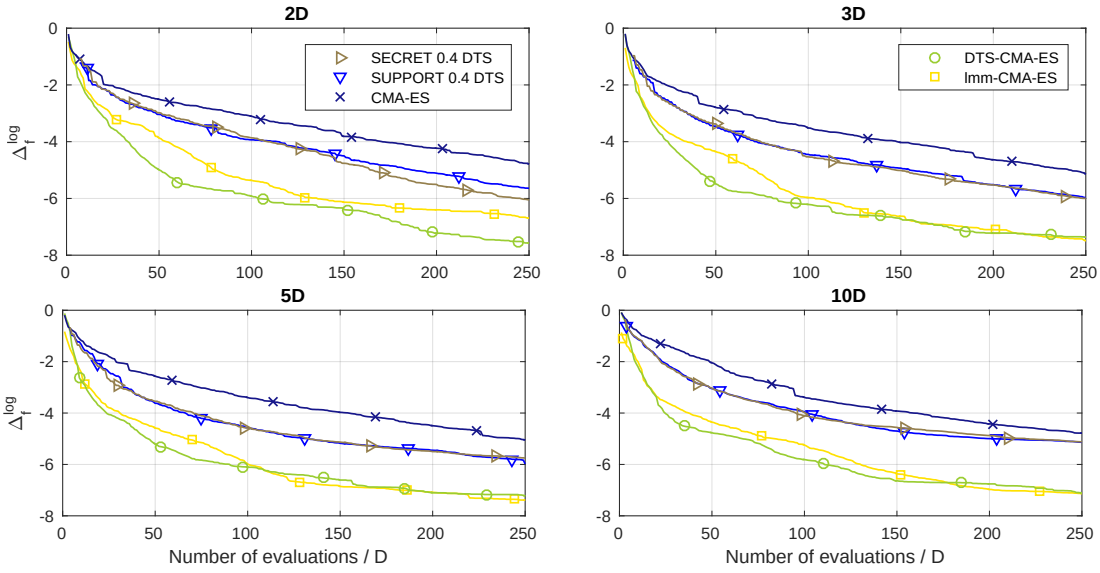


Figure 36: Scaled median distances Δ_f^{log} averaged over all 24 COCO functions in 2D, 3D, 5D, and 10D for algorithms CMA-ES, DTS-CMA-ES, lmm-CMA, and 2 RF settings of DTS-CMA-ES.

used to predict the values of the true objective function. Therefore, it is very difficult to understand what impact on the performance of a particular surrogate modelling method has the choice of the employed regression model, and what impact has its EC. To contribute to such understanding was the aim of the research reported in (Pitra et al., 2021).

Among the most successful of CMA-ES surrogate variants, we have selected three that we view, according to their published descriptions, as paying most attention to EC: the lmm-CMA (Kern et al., 2006; Auger et al., 2013), the DTS-CMA-ES (Pitra et al., 2016; Bajer et al., 2019), and the lq-CMA-ES (Hansen, 2019) (see Sections 2.3.3 and 4.4). We have analysed what exactly constitutes the EC of each of those methods, relying apart from the published descriptions also on their publicly available implementations. That allowed us to subsequently implement all possible combinations of the regression models employed in them with the three specific ECs, and to compare the models and their EC separately. The experiments were performed on the noiseless and noisy benchmarks of the COCO platform and a real-world simulation benchmark by Wu et al. (2016).

4.8.1 *Analysis of the EC in Important Surrogate-Assisted CMA-ES Variants*

In this section, three algorithms that we consider most important among the surrogate-assisted CMA-ES variants will be analysed. Each of them has been originally proposed as a whole, addressing simultaneously both components of surrogate modelling:

1. The employed model, the purpose of which is to construct a regression approximation of the true black-box objective function (bb).
2. The EC, the purpose of which is to select points in which points the bb should be evaluated, and in which using the regression approximation is sufficient.

We analysed which parts of the algorithm concern the employed model and which concern its EC. That analysis enabled us an implementation of all combinations of the three models employed in the described algorithms with the three specific ECs for an experimental investigation of interactions between the model and its EC.

The analysis is facilitated by the fact that all three algorithms share several key properties and concepts. In particular, the population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda \in \mathbb{R}^D$, the parent number μ, the concept of an archive $\mathcal{A}$ of points evaluated by the bb, and the concept of a ranking function $\rho : \mathbb{R}^m \to \Pi(m)$, with $\Pi(m)$ denoting the set of permutations of $\{1, \ldots, m\}$, e. g., in case of increasing ranking $(\forall \mathbf{y} \in \mathbb{R}^m)(\rho(\mathbf{y}))_i < (\rho(\mathbf{y}))_j \Rightarrow y_i \leq y_j$.

Algorithm lmm-CMA

In the local-metamodel-CMA, a specific full quadratic model $f_j : \mathbb{R}^D \to \mathbb{R}$ is trained for $\mathbf{x}_j, j = 1, \ldots, \lambda$ (Auger et al., 2013),

$$f_j(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_j)^\top \mathbf{A}_j (\mathbf{x} - \mathbf{x}_j) + (\mathbf{x} - \mathbf{x}_j)^\top \mathbf{b}_j + c_j \text{ with } \mathbf{A}_j \in \mathbb{R}^{D \times D}, \mathbf{b}_j \in \mathbb{R}^D, c_j \in \mathbb{R}\,. \tag{96}$$

The model f_j is trained on the set $N_k(\mathbf{x}_j; \mathcal{A})$ of a given number k of nearest neighbours of $\mathbf{x}_j$ with respect to a given archive $\mathcal{A}$,

$$N_k(\mathbf{x}_j; \mathcal{A}) \subset \mathcal{A}, |N_k(\mathbf{x}_j; \mathcal{A})| = k\,, (\forall \mathbf{x} \in N_k(\mathbf{x}_j; \mathcal{A}))(\forall \mathbf{x}' \in \mathcal{A} \setminus N_k(\mathbf{x}_j; \mathcal{A}))\, d(\mathbf{x}, \mathbf{x}_j) \leq d(\mathbf{x}', \mathbf{x}_j)\,. \tag{97}$$

As the distance d in (97), the Mahalanobis distance for $\sigma^2\mathbf{C}$ is used:

$$d_{\sigma^2\mathbf{C}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \sigma^{-2} \mathbf{C}^{-1} (\mathbf{x} - \mathbf{y})} = \|\mathbf{M}(\mathbf{x} - \mathbf{y})\|\,, \text{ with } \mathbf{M} = \frac{1}{\sigma} \mathbf{D}^{-\frac{1}{2}} \mathbf{B}^\top\,, \tag{98}$$

Algorithm 13 Using the surrogate model in lmm-CMA

Input: population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda \in \mathbb{R}^D$, number of parents μ, archive $\mathcal{A}$ with $|\mathcal{A}| \geq D(D+3)/2+2$

1: Set $\epsilon_0 = +\infty$, $\epsilon_{\max} = \lambda^2/20$, $i = 0$, $k = \min(D(D+3)+2, \lceil\sqrt{|\mathcal{A}|(D(D+3)/2+1)}\rceil)$, $n_{\text{iter}} = \lceil\lambda/20\rceil$, $n_{\text{init}} = 1$ (if the model is used 1st time in the current CMA-ES run)
2: Train f_j on $N_k(\mathbf{x}_j; \mathcal{A})$ for $j = 1, \ldots, \lambda$ according to (96)
3: Set $\mathcal{P} = \{\mathbf{x}_j | (\rho(f_1(\mathbf{x}_1), \ldots, f_\lambda(\mathbf{x}_\lambda)))_j > n_{\text{init}}\}$
4: Evaluate $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$ not yet $\mathfrak{bb}$-evaluated
5: Update $\mathcal{A}$ to $\mathcal{A} \cup \{\mathbf{x}_j | (\rho(f_1(\mathbf{x}_1), \ldots, f_\lambda(\mathbf{x}_\lambda)))_j \leq n_{\text{init}}\}$
6: Update $N_k(\mathbf{x}_j; \mathcal{A})$ according to (97)
7: Train f_j on $N_k(\mathbf{x}_j; \mathcal{A})$ for $j \in \mathcal{P}$ according to (96)
8: Define $f^0(\mathbf{x}_j) = f_j(\mathbf{x}_j)$ for $\mathbf{x}_j \in \mathcal{P}$, $f^0(\mathbf{x}_j) = \mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$
9: **while** $\mathcal{P}_{\neg\mathfrak{bb}} \neq \emptyset$ & $\epsilon_i > \epsilon_{\max}$ **do**
10: Set $i = i + 1$ and update $\mathcal{P}$ to $\{\mathbf{x}_j^{\mathcal{P}} | (\rho(f(\mathbf{x}^{\mathcal{P}}))_j > n_{\text{iter}}\}$
11: Evaluate $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$ not yet $\mathfrak{bb}$-evaluated
12: Update $\mathcal{A}$ to $\mathcal{A} \cup \{\mathbf{x}_j | \rho(f(\mathbf{x}^{\mathcal{P}}))_j \leq n_{\text{iter}}\}$
13: Update $N_k(\mathbf{x}_j; \mathcal{A})$ according to (97)
14: Train f_j on $N_k(\mathbf{x}_j; \mathcal{A})$ for $\mathbf{x}_j \in \mathcal{P}$ according to (96)
15: Define $f^i(\mathbf{x}_j) = f_j(\mathbf{x}_j)$ for $\mathbf{x}_j \in \mathcal{P}$, $f^i(\mathbf{x}_j) = \mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$
16: $\epsilon_i = \sum_{j:(\rho(f^i))_j \leq \mu} |(\rho(f^i)_j - \rho(f^{i-1}))_j|$
17: **end while**
18: Update $f^i(\mathbf{x}_j)$ to $f^i(\mathbf{x}_j) - \min_{\mathbf{x}_{j'} \in \mathcal{P}} f^i(\mathbf{x}_{j'}) + \min_{\mathbf{x}_{j'} \notin \mathcal{P}} f^i(\mathbf{x}_{j'})$, $\mathbf{x}_j \in \mathcal{P}$
19: Update $n_{\text{init}} = \begin{cases} \max(0, n_{\text{init}} - n_{\text{iter}}) & \text{for } i = 1 \\ \min(\lambda, n_{\text{init}} + n_{\text{iter}}) & \text{for } i \geq 3 \end{cases}$

Output: $f^i(\mathbf{x}_1), \ldots, f^i(\mathbf{x}_\lambda), n_{\text{init}}$

where $\|\cdot\|$ denotes the Euclidean norm, and $\mathbf{M}$ is the matrix of the linear Mahalanobis decorrelation transformation for $\sigma^2\mathbf{C}$, in which $\mathbf{D}$ denotes the diagonal matrix of the eigenvalues of $\mathbf{C}$ and $\mathbf{B}$ the matrix of its orthonormal eigenvectors. Due to (98), it is advantageous to express (96) in terms of Mahalanobis-transformed $\mathbf{x} - \mathbf{x}_j$,

$$f_j(\mathbf{x}) = (\mathbf{M}(\mathbf{x} - \mathbf{x}_j))^\top \mathbf{A}_j^{\mathbf{M}} (\mathbf{M}(\mathbf{x} - \mathbf{x}_j)) + (\mathbf{M}(\mathbf{x} - \mathbf{x}_j))^\top \mathbf{b}_j^{\mathbf{M}}) + c_j\,, \tag{99}$$

which corresponds to $\mathbf{A}_j = \mathbf{M}^\top \mathbf{A}_j^{\mathbf{M}} \mathbf{M}$ and $\mathbf{b}_j = \mathbf{M}^\top \mathbf{b}_j^{\mathbf{M}}$.

The algorithm lmm-CMA employs $f_1, \ldots, f_\lambda$ to find an evaluation of the population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$, as described in Algorithm 13. It uses the notation

$$\mathcal{P}_{\neg\mathfrak{bb}} = \{\mathbf{x}_j | j = 1, \ldots, \lambda \;\&\; \mathbf{x}_j \text{ has not yet been evaluated by } \mathfrak{bb}\}$$

and for a population subsets $\mathcal{P}$, the notation

$$\mathcal{P} = \{\mathbf{x}_1^{\mathcal{P}}, \ldots, \mathbf{x}_{|\mathcal{P}|}^{\mathcal{P}}\}\,, \quad f(\mathbf{x}^{\mathcal{P}}) = (f_1(\mathbf{x}_1^{\mathcal{P}}), \ldots, f_{|\mathcal{P}|}(\mathbf{x}_{|\mathcal{P}|}^{\mathcal{P}}))\,.$$

Observe that all points are first evaluated by a model, and the decisions which of them to evaluate by the true objective function, are made in the lines 4, 8, 11, and 15, depending on the line 19. Hence, we will take the lines 4, 8, 11, 15, and 19 as concerning the EC, and the remaining parts of the algorithm as concerning the employed model.

Algorithm 14 Using the surrogate model in DTS-CMA-ES

Input: population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda \in \mathbb{R}^D$, number of parents μ, archive $\mathcal{A}$, step-size σ and matrix $\mathbf{C}$ from the CMA-ES distribution, $\mathfrak{C}$ – selection criterion for evaluation by the $\mathfrak{bb}$, radius $r_{\max} > 0$, self-adaptation parametres $\beta, \epsilon_{\min}, \epsilon_{\max}, \alpha_{\min}, \alpha_{\max} \in (0, 1)$

1: Set $\alpha = \epsilon = 0.05$ for the 1st model usage in the current CMA-ES run
2: Train a GP f_1 on $\mathcal{T}_{k_\mathcal{A}}$, estimating hyperparameters through MLE (15)
3: Evaluate $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j$ not yet $\mathfrak{bb}$-evaluated and such that $(\rho(\mathfrak{C}(\mathbf{x}_1), \ldots, \mathfrak{C}(\mathbf{x}_\lambda)))_j \leq \lceil \alpha\lambda \rceil$
4: Update $\mathcal{A}$ to $\mathcal{A} \cup \{\mathbf{x}_j | (\rho(\mathfrak{C}(\mathbf{x}))_j \leq \lceil \alpha\lambda \rceil\}$
5: Train a GP f_2 on $\mathcal{T}_{k_\mathcal{A}}$, estimating hyperparameters through MLE (15)
6: Update $f_2(\mathbf{x}_j)$ to $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j$ such that $\rho(\mathfrak{C}(\mathbf{x}))_j \leq \lceil \alpha\lambda \rceil$
7: Update ϵ to $(1 - \beta)\epsilon + \beta \mathrm{err}_{\mathrm{RDE}}(f_1(\mathbf{x}), f_2(\mathbf{x}))$
8: Update α to $\alpha_{\min} + \max(0, \min(1, \frac{\epsilon - \epsilon_{\min}}{\epsilon_{\max} - \epsilon_{\min}}))$
9: Update $f_2(\mathbf{x}_j)$ to $f_2(\mathbf{x}_j) - \min\{f_2(\mathbf{x}_{j'}) | \rho(\mathfrak{C}(\mathbf{x}))_{j'} > \lceil \alpha\lambda \rceil\} + \min\{f_2(\mathbf{x}_{j'}) | \rho(\mathfrak{C}(\mathbf{x}))_{j'} \leq \lceil \alpha\lambda \rceil\}$ for j fulfilling $\rho(\mathfrak{C}(\mathbf{x}))_j > \lceil \alpha\lambda \rceil$

Output: $f_2(\mathbf{x}_1), \ldots, f_2(\mathbf{x}_\lambda), \epsilon, \alpha$

Algorithm DTS-CMA-ES

The surrogate model in the DTS-CMA-ES (Bajer et al., 2019) is a GP using $\kappa^{5/2}_{\mathrm{Mat}}$ covariance function and according to a comparison of three such covariance functions published in (Bajer et al., 2019) (see Section 4.4.2), it led to the best performance of the DTS-CMA-ES. Due to the invariance of the CMA-ES with respect to monotonous transformations, that performance was measured using the RDE of GP predictions with respect to values of the true objective function, considering μ best components (see Equation (67)). For the selection of $\mathfrak{bb}$-evaluated points, we employed two criteria: $\mathfrak{C}_{\mathrm{M}}$ and $\mathfrak{C}_{\mathrm{PoI}}$ (see Equations (72) and (74) respectively). To have an ordering consistent with the function values criterion employed in the algorithms lmm-CMA and lq-CMA-ES, we used $-\mathfrak{C}_{\mathrm{M}}$ and $1 - \mathfrak{C}_{\mathrm{PoI}}$ instead which entails an increasing ordering, i. e., the first value is the lowest.

The algorithm DTS-CMA-ES employs two consecutively trained GPs to find an evaluation of the population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$, as described in Algorithm 14. It uses the notation $f_i(\mathbf{x}) = (f_i(\mathbf{x}_1), \ldots, f_i(\mathbf{x}_\lambda))$, $i = 1, 2$, $\mathfrak{C}(\mathbf{x}) = (\mathfrak{C}(\mathbf{x}_1), \ldots, \mathfrak{C}(\mathbf{x}_j))$ for an ordering $\mathfrak{C}$, and $k_\mathcal{A} = \max\{h || \mathcal{T}_h | \leq N_{\max}\}$, where $N_{\max} \in \mathbb{N}, N_{\max} \geq \lambda$ and

$$\mathcal{T}_h = \bigcup_{j=1}^{\lambda} \{\mathbf{x} \in N_h(\mathbf{x}_j; \mathcal{A}) | d_{\sigma^2\mathbf{C}}(\mathbf{x}, \mathbf{x}_j) < r_{\max}\} \text{ for } h = 1, \ldots, |\mathcal{A}|\,. \tag{100}$$

The decisions which points should be evaluated by the true objective function and which by the GP f_2, are made in the lines 3 and 6 depending on adjustments in the lines 1 and 7. Therefore, we will take the lines 1, 3, 6 and 7 as concerning the EC, and the remaining parts of the algorithm as concerning the employed model.

Algorithm lq-CMA-ES

The algorithm lq-CMA-ES is similar to the lmm-CMA in employing a quadratic surrogate model and in repeatedly evaluating within a single CMA-ES generation additional parts of the population with the $\mathfrak{bb}$ unless the performance of the current surrogate model is satisfactory. However, it differs from the lmm-CMA in several important respects:

1. Whereas surrogate models in the lmm-CMA are always full quadratic, the lq-CMA-ES admits also pure quadratic or linear models. The employed kind of model depends on the number of $\mathfrak{bb}$-evaluated points available for model training, which gradually increases if the performance of the surrogate model is not satisfactory. More precisely, if k $\mathfrak{bb}$-evaluated points are available to train the respective model f_k, then the kind of f_k is
   - ◇ full quadratic if $k/1.1 \geq D(D+3)/2+1$,
   - ◇ pure quadratic if $2D+1 \leq k/1.1 < D(D+3)/2+1$,
   - ◇ linear if $D+1 \leq k/1.1 < 2D+1$.
2. Training points are taken from a queue $Q \subset \mathcal{A}$. New points and their $\mathfrak{bb}$-evaluations are appended to its end, causing points at its beginning to be dropped if its length would otherwise exceed a given limit.
3. $\mathfrak{bb}$-evaluations are used to assess the performance of the surrogate model, by means of the Kendall's rank correlation coefficient τ between those evaluations and model predictions. They are not returned to the CMA-ES unless no satisfactory model was found before the whole population was $\mathfrak{bb}$-evaluated, then the $\mathfrak{bb}$ values are returned for the whole population. Otherwise, predictions by the model are returned instead, even for points that have been $\mathfrak{bb}$-evaluated.

How the algorithm lq-CMA-ES finds an evaluation of the population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ is described in Algorithm 15. It uses the notation $\mathbf{y}_1, \ldots, \mathbf{y}_{|Q|}$ for the elements of Q in its current ordering and the notation

$$\tau_{k^Q} = \tau((f_{k^Q}(\mathbf{y}_j))_{j=|Q|-L_k+1}^{k^Q}, (\mathfrak{bb}(\mathbf{y}_j))_{j=|Q|-L_k+1}^{k^Q}), \tag{101}$$

where L_k is the number of last elements of Q used to assess with τ the performance of f_k and it is defined in (Hansen, 2019) as $L_k = \max(15, \min(1.2k, 0.75\lambda))$.

The decisions which points should be evaluated by the true objective function and which by the f_{k^Q}, are made in the lines 4, 13, and 19–23, and they depend on decisions made in the lines 5–7 and 14–16. Therefore, we will take the lines 4–7, 13–16, and 19–23 as concerning the EC, and the remaining parts of the algorithm as concerning the employed model.

Using an Evolution Control for Another Model

Observe that in all three above algorithms, the EC interacts with the population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ evaluated by the model by means of some ordering:

1. In the algorithm lmm-CMA, the population or its subset $\mathcal{P}$ is ordered according to the ordering $f_j(\mathbf{x}_j)$.
2. In the algorithm DTS-CMA-ES, the population is ordered according to the ordering $\rho_{\mathfrak{C}}(\mathbf{x}_j)$, considering a criterion $\mathfrak{C}$ for the selection of points to be evaluated by the $\mathfrak{bb}$.
3. In the algorithm lq-CMA-ES, the population or its subset $\mathcal{P}$ is ordered according to the ordering $f_{k^Q}(\mathbf{x}_j)$.

This suggests a straightforward way how to combine the EC of one of these three algorithms (Algorithm A) with the model employed in another (Algorithm B): to evaluate the population

Algorithm 15 Using the surrogate model in lq-CMA-ES

Input: population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda \in \mathbb{R}^D$, archive $\mathcal{A}$ fulfilling $|\mathcal{A}| \geq 1.1(D+1)$, queue $Q \subset \mathcal{A}$

Set $k^Q = \lfloor 1 + \max(0.02\lambda, 4 - |Q|) \rfloor$
Train f_{k^Q} on the k^Q last points from Q
Set $\mathcal{P} = \{\mathbf{x}_j | (\rho(f_{k^Q}(\mathbf{x}_1), \ldots, f_{k^Q}(\mathbf{x}_\lambda)))_j > k^Q\}$
Evaluate $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$ not yet $\mathfrak{bb}$-evaluated and append those $\mathbf{x}_j$ to Q
if $|Q| > \max(\lambda, D(D+3)+2)$ **then**
  Drop first $|Q| - \max(\lambda, D(D+3)+2)$ elements from Q
end if
Reorder the last $\min(k^Q, \lambda)$ elements of Q decreasingly with respect to their $\mathfrak{bb}$ values
while $\mathcal{P} \neq \emptyset$ & $\tau_{k^Q} < 0.85$ **do**
  Update k^Q to $1.5k^Q$
  Train f_{k^Q} on the k^Q last points from Q
  Update $\mathcal{P}$ to $\{\mathbf{x}_j | (\rho(f_{k^Q}(\mathbf{x}_1^{\mathcal{P}}), \ldots, f_{k^Q}(\mathbf{x}_{|\mathcal{P}|}^{\mathcal{P}})))_j > k^Q + |\mathcal{P}| - \lambda\}$
  Evaluate $\mathfrak{bb}(\mathbf{x}_j)$ for $\mathbf{x}_j \notin \mathcal{P}$ not yet $\mathfrak{bb}$-evaluated and append those $\mathbf{x}_j$ to Q
  if $|Q| > \max(\lambda, D(D+3)+2)$ **then**
    Drop first $|Q| - \max(\lambda, D(D+3)+2)$ elements from Q
  end if
  Reorder the last $\min(k^Q, \lambda)$ elements of Q decreasingly with respect to their $\mathfrak{bb}$ values
end while
if $\mathcal{P} = \emptyset$ **then**
  Return Q, $\mathfrak{bb}(\mathbf{x}_1), \ldots, \mathfrak{bb}(\mathbf{x}_\lambda)$
else
  Return Q, $f_{k^Q}(\mathbf{x}_j) - \min_{j'=1}^{\lambda} f_{k^Q}(\mathbf{x}_{j'}) + \min_{j'=1}^{\lambda} \mathfrak{bb}(\mathbf{x}_{j'}), j = 1, \ldots, \lambda$
end if

Output: Q, model- or $\mathfrak{bb}$-evaluated population $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$

or its subset $\mathcal{P}$ by the model employed in Algorithm B and to order it by the ordering considered in Algorithm B instead of the ordering originally considered in Algorithm A. We used this approach to implement all possible combinations of the models employed in the above three algorithms with the EC strategies encountered in them. Moreover, the GP model employed in DTS-CMA-ES was used with the two criteria $\mathfrak{C}_{\mathrm{M}}$ and $\mathfrak{C}_{\mathrm{PoI}}$. Hence, altogether 12 such combinations have been implemented.

4.8.2 *Experimental Investigation of Interactions between Model and Its Control*

The core of this section lies in a systematic comparison of the three mentioned approaches to surrogate model usage with the CMA-ES in combination with four surrogate models. We have tested the EC methods from the lmm-CMA, DTS-CMA-ES, and lq-CMA-ES in combination with the locally-weighted quadratic models, GP using function values, GP using PoI, and global linear-quadratic models.

Experimental Set-up

We have performed experimental evaluation on the 24 noiseless f_{1-24} and 30 noisy $f_{101-130}$ single-objective COCO benchmarks (Hansen et al., 2009a,b, 2012) in dimensions 2, 3, 5, 10, and 20, considering always 15 instances of each function. Moreover, we have also evaluated the benchmark f_{sim} simulating energy landscape based on a layout of wave energy converters presented by Wu et al. (2016) for 1, 2, 5, 6, 8, and 10 buoys, i. e., in dimensions 2, 4, 10, 12, 16, and 20, on all 24 combinations of the following simulation settings: simulation frequencies 1 and 2, buoy radiuses 2.0, 2.5, 3.2, and search space sizes $[10, 10]$, $[25, 25]$, $[50, 50]$, and $[100, 100]$. Each algorithm had a budget of $250\,\text{FE}/D$ to reach the target distance $\Delta f_T = 10^{-8}$ from the function optimum. The parameters of the tested algorithms are summarized in the following paragraphs.

BASE CMA-ES The base CMA-ES identical for all surrogate-assisted combinations was employed in its IPOP-CMA-ES version (Matlab code v. 3.62β) with the following settings: number of restarts $= 50$, IncPopSize $= 2$, $\sigma^{(0)} = \frac{8}{3}$, $\lambda = 8 + \lfloor 6 \ln D \rfloor$, initial point $\mathbf{m}^{(0)} \sim \mathcal{U}[-4, 4]^D$. The remaining settings were left default.

LMM-CMA The lmm-CMA evolution control and locally-weighted quadratic regression model were used in its improved version published by Auger et al. (2013).

DTS-CMA-ES The DTS-CMA-ES evolution control and the GP model were utilized in the default settings of its adaptive version (Bajer et al., 2019).

LQ-CMA-ES The lq-CMA-ES evolution control and linear-quadratic model were used in the version published by Hansen (2019).

We use a budget-dependent quality measure of the compared algorithms, strongly influenced by the concept of target precision values used in the COCO framework (Hansen et al., 2021). In particular, we measure the subset of target precision values achievable by the assessed algorithm at the considered budget within an apriori selected set of target precision values. To select a set incorporating all possible target values of tested benchmarks, we have symmetrically doubled on the logarithmic scale the default set $[10^{-8}, 10^{2}]$ utilized in (Hansen et al., 2021), which yields the interval $[10^{-13}, 10^{7}]$. The subset of achievable target precision values is measured with a measure that we denote Δ_f^{μ}: the Lebesgue measure of the logarithmic transformation of this set, normalized to yield the maximal value 1 if all target precision values from $[10^{-13}, 10^{7}]$ are achievable. We consider two evaluation budgets ($\frac{1}{5}$ of the full budget and the full budget, i. e., $50\,\text{FE}/D$ and $250\,\text{FE}/D$) and 12 groups of functions (f_{1-5}, f_{6-9}, f_{10-14}, f_{15-19}, f_{20-24}, f_{1-24}, $f_{101-106}$, $f_{107-121}$, $f_{122-130}$, $f_{101-130}$, f_{sim}, and $f_{1-24,101-130,\text{sim}}$).

Investigation on the Noiseless COCO Benchmarks

Results from experiments on the noiseless part of the COCO platform Hansen et al. (2009b) are presented in Figures 1–36 and in Tables 1–36 in online available SUPPLEMENTARY MATERIAL[13]. In the paper, we report only summary of the results in Table 24 and the performances on few selected functions in Figure 37. In this Figure, we show the dependence of the scaled best-

13 https://raw.githubusercontent.com/jdgregorian/surrogate-cmaes-pages/gh-pages/supp/gecco2021/supp_mat.pdf

Table 24: A pairwise comparison of the evolution controls, models, and their combinations in $2D$, $3D$, $5D$, $10D$ and $20D$ over the noiseless COCO benchmarks for different evaluation budgets. The percentage of wins of i-th algorithm against j-th algorithm over all benchmark instances is given in the i-th row and j-th column. The numbers in bold mark the row algorithm being significantly better than the column algorithm according to the two-sided Wilcoxon signed rank test with the Holm correction at family-wise significance level $\alpha = 0.05$.

$\mathbf{2-20}D$	$\mathrm{DT_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$		$\mathrm{DT_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$		$\mathrm{DT_{EC}}+\mathrm{lq_{\mathcal{M}}}$		$\mathrm{DT_{EC}}+\mathrm{lmm_{\mathcal{M}}}$		$\mathrm{lq_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$		$\mathrm{lq_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$		$\mathrm{lq_{EC}}+\mathrm{lq_{\mathcal{M}}}$		$\mathrm{lq_{EC}}+\mathrm{lmm_{\mathcal{M}}}$		$\mathrm{lmm_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$		$\mathrm{lmm_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$		$\mathrm{lmm_{EC}}+\mathrm{lq_{\mathcal{M}}}$		$\mathrm{lmm_{EC}}+\mathrm{lmm_{\mathcal{M}}}$	
#FEs/D	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250
$\mathrm{DT_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$	—	—	33	44	**66**	**58**	44	54	43	**55**	39	52	48	47	53	**55**	**59**	**54**	**55**	47	**67**	**66**	**64**	49
$\mathrm{DT_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$	**67**	**56**	—	—	**77**	**62**	**59**	**58**	**60**	**63**	**54**	**59**	**63**	51	**69**	**60**	**72**	**60**	**70**	53	**77**	**67**	**74**	**55**
$\mathrm{DT_{EC}}+\mathrm{lq_{\mathcal{M}}}$	35	42	23	39	—	—	30	48	34	49.7	27	46	32	39	37	49	44	49.6	40	43	51	**62**	45	45
$\mathrm{DT_{EC}}+\mathrm{lmm_{\mathcal{M}}}$	**56**	46	41	42	**70**	**52**	—	—	50	**53**	45	49	**52**	44	**61**	**53**	**62**	**51**	**61**	44	**73**	**59**	**69**	45
$\mathrm{lq_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$	**57**	45	40	37	**66**	50.3	50	47	—	—	45	45	53	40	**57**	47	**62**	47	**59**	41	**67**	**56**	**64**	42
$\mathrm{lq_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$	**61**	48	46	41	**73**	**54**	**55**	51	**55**	**55**	—	—	**58**	44	**61**	51	**68**	52	**65**	45	**72**	**59**	**68**	46
$\mathrm{lq_{EC}}+\mathrm{lq_{\mathcal{M}}}$	**52**	**53**	37	49	**68**	**61**	48	**56**	47	**60**	42	**56**	—	—	**56**	**61**	**60**	**58**	**58**	**52**	**71**	**70**	**64**	**54**
$\mathrm{lq_{EC}}+\mathrm{lmm_{\mathcal{M}}}$	47	45	31	40	**63**	51	39	47	43	**53**	39	50	44	39	—	—	**56**	**51**	**54**	46	**65**	**59**	**63**	46
$\mathrm{lmm_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$	41	46	28	40	**56**	50.4	38	49	38	**53**	32	48	40	42	44	49	—	—	49.7	41	**60**	**60**	**58**	46
$\mathrm{lmm_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$	45	53	30	47	**60**	**57**	39	56	41	**59**	35	**55**	42	48	46	54	50.3	**60**	—	—	**60**	**67**	**57**	53
$\mathrm{lmm_{EC}}+\mathrm{lq_{\mathcal{M}}}$	33	34	23	33	49	39	27	41	33	44	28	41	29	30	35	41	40	40	40	33	—	—	47	36
$\mathrm{lmm_{EC}}+\mathrm{lmm_{\mathcal{M}}}$	36	51	26	45	**55**	**55**	31	55	36	**58**	32	**54**	36	46	37	**54**	42	**54**	43	47	**53**	**64**	—	—

$\mathbf{2-20}D$	$\mathrm{DT_{EC}}$		$\mathrm{lq_{EC}}$		$\mathrm{lmm_{EC}}$	
#FEs/D	50	250	50	250	50	250
$\mathrm{DT_{EC}}$	—	—	46	49.9	**61**	**53**
$\mathrm{lq_{EC}}$	**54**	50.1	—	—	**65**	**52**
$\mathrm{lmm_{EC}}$	39	47	35	48	—	—

$\mathbf{2-20}D$	$\mathrm{GP^{M}_{\mathcal{M}}}$		$\mathrm{GP^{PoI}_{\mathcal{M}}}$		$\mathrm{lq_{\mathcal{M}}}$		$\mathrm{lmm_{\mathcal{M}}}$	
#FEs/D	50	250	50	250	50	250	50	250
$\mathrm{GP^{M}_{\mathcal{M}}}$	—	—	45	43	**59**	53	53	49
$\mathrm{GP^{PoI}_{\mathcal{M}}}$	**55**	**57**	—	—	**61**	**57**	**59**	**53**
$\mathrm{lq_{\mathcal{M}}}$	41	47	39	43	—	—	44	48
$\mathrm{lmm_{\mathcal{M}}}$	47	**51**	41	47	**56**	**52**	—	—

achieved logarithms $\Delta_f^{\log}$ of median distances Δ_f^{med} to the optimal fitness value on the number of fitness evaluations divided by the dimension. Medians Δ_f^{med}, 1^{st}, and 3^{rd} quartiles are calculated from 15 (24 for f_{sim}) independent instances for each respective algorithm, function, and dimension (see Section 2.6.2 for more detailed explanation of convergence graphs).

We tested pairwise differences in performance measured with Δ_f^{μ} for all EC methods (denoted according to the original algorithm $\mathrm{lmm_{EC}}$, $\mathrm{DT_{EC}}$, $\mathrm{lq_{EC}}$), surrogate models (denoted $\mathrm{lmm_{\mathcal{M}}}$, $\mathrm{GP^{M}_{\mathcal{M}}}$, $\mathrm{GP^{PoI}_{\mathcal{M}}}$, $\mathrm{lq_{\mathcal{M}}}$), and their combinations on the noisy and noiseless part of COCO framework and the simulation benchmark using the non-parametric two-sided Wilcoxon signed rank test with the Holm correction for the family-wise error. To better illustrate the differences between individual settings, we also count the percentage of instances at which one combination/EC/model had the Δ_f^{μ} higher than the other. The pairwise score and the statistical significance of the pairwise differences are summarized in Table 24.

The two most successful combinations on the noiseless functions were $\mathrm{DT_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$ and $\mathrm{lq_{EC}}+\mathrm{lq_{\mathcal{M}}}$. Other ECs in combination with $\mathrm{lq_{\mathcal{M}}}$ possibly could not handle the simplicity of this model and provided the worst performance among all compared combinations except few functions in $20D$, e. g., on the Attractive sector function f_6. The performance of combinations during the optimization of Attractive sector function f_6 scales differently for different combinations. For example, the $\mathrm{DT_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$ improves from being the worst combination in $2D$ to being the best in $20D$, and also the performance of $\mathrm{lmm_{EC}}$ combinations improves with dimension, whereas the performance of the $\mathrm{DT_{EC}}+\mathrm{lmm_{\mathcal{M}}}$, $\mathrm{DT_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$, $\mathrm{lq_{EC}}+\mathrm{GP^{M}_{\mathcal{M}}}$, and $\mathrm{DT_{EC}}+\mathrm{GP^{PoI}_{\mathcal{M}}}$ on f_6 decreases with dimension. The $\mathrm{lq_{EC}}$ in combination with GP models

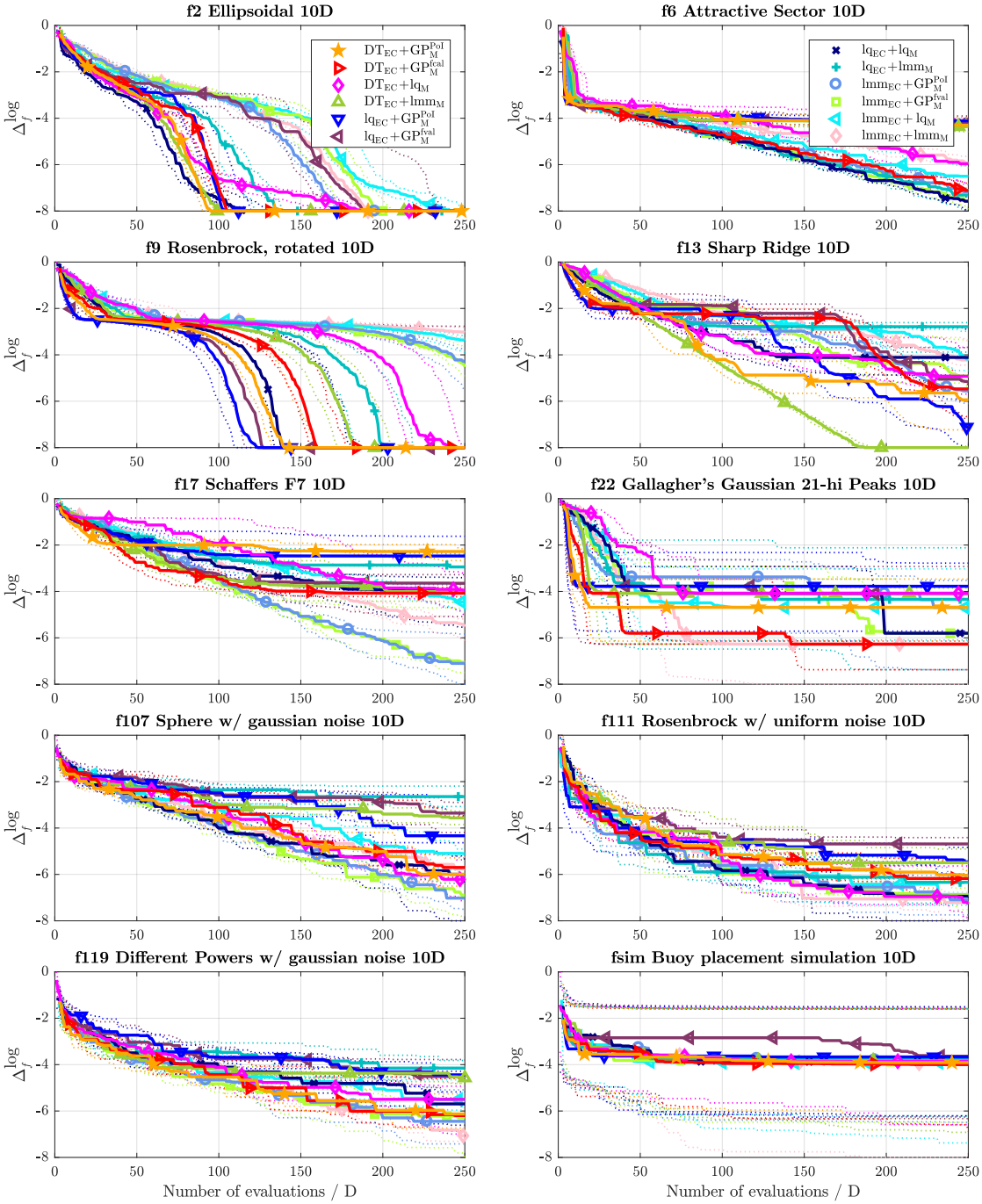


Figure 37: Medians (solid) and $1^{st}/3^{rd}$ quartiles (dotted) of the distances to the optima of 10 function selected from 24 noiseless, 30 noisy COCO, and the simulation benchmark in 10D for all compared EC — model combinations. The $\log_{10}$ of medians/quartiles were calculated across 15 (24 for f_{sim}) independent function instances for each combination and linearly scaled to $[-8, 0]$.

Table 25: A pairwise comparison of the evolution controls, models, and their combinations in $2D$, $3D$, $5D$, $10D$ and $20D$ over the noisy COCO benchmarks for different evaluation budgets. The percentage of wins of i-th algorithm against j-th algorithm over all benchmark instances is given in the i-th row and j-th column. The numbers in bold mark the row algorithm being significantly better than the column algorithm according to the two-sided Wilcoxon signed rank test with the Holm correction at family-wise significance level $\alpha = 0.05$.

$\mathbf{2-20}D$	$\mathrm{DT_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{DT_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{DT_{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{DT_{EC}}+\mathrm{lmm}_{\mathcal{M}}$		$\mathrm{lq_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{lq_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lq_{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lq_{EC}}+\mathrm{lmm}_{\mathcal{M}}$		$\mathrm{lmm_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{lmm_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lmm_{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lmm_{EC}}+\mathrm{lmm}_{\mathcal{M}}$	
#FEs/D	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250
$\mathrm{DT_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	—	—	50.1	48	**58**	46	49	**59**	**54**	**60**	52	**58**	48	41	53	52	52	**56**	49.7	51	**60**	**53**	**54**	**52**
$\mathrm{DT_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	49.9	**52**	—	—	**59**	49	49	**62**	**54**	**61**	**52**	**61**	49.7	45	**54**	**55**	53	**58**	52	**53**	**60**	**56**	**54**	**55**
$\mathrm{DT_{EC}}+\mathrm{lq}_{\mathcal{M}}$	42	**54**	41	51	—	—	42	**61**	48	**65**	46	**62**	40	47	48	**56**	45	**60**	43	**56**	51	**58**	46	**58**
$\mathrm{DT_{EC}}+\mathrm{lmm}_{\mathcal{M}}$	51	41	51	38	**58**	39	—	—	**55**	51	**54**	49	48	34	**56**	44	**53**	49	52	43	**59**	43	**56**	45
$\mathrm{lq_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	46	40	46	39	52	35	45	49	—	—	47	49.8	44	34	49	45	48	47	48	41	54	43	49	42
$\mathrm{lq_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	48	42	48	39	54	38	46	51	53	50.2	—	—	46	34	52	46	51	48	49	43	56	43	51	44
$\mathrm{lq_{EC}}+\mathrm{lq}_{\mathcal{M}}$	52	**59**	50.3	**55**	**60**	**53**	52	**66**	**56**	**66**	**54**	**66**	—	—	**56**	**59**	**55**	**62**	**53**	**58**	**61**	**61**	**57**	**59**
$\mathrm{lq_{EC}}+\mathrm{lmm}_{\mathcal{M}}$	47	48	46	45	52	44	44	**56**	51	**55**	48	**54**	44	41	—	—	49	**53**	48	49	**55**	49	51	50.1
$\mathrm{lmm_{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	48	44	47	42	**55**	40	47	51	52	**53**	49	52	45	38	51	47	—	—	49.5	45	**56**	46	**54**	48
$\mathrm{lmm_{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	50.3	49	48	47	**57**	44	48	**57**	**52**	**59**	51	**57**	47	42	52	51	50.5	**55**	—	—	**58**	51	**55**	52
$\mathrm{lmm_{EC}}+\mathrm{lq}_{\mathcal{M}}$	40	47	40	44	49	42	41	**57**	46	**57**	44	**57**	39	39	45	51	44	**54**	42	49	—	—	46	49.7
$\mathrm{lmm_{EC}}+\mathrm{lmm}_{\mathcal{M}}$	46	48	46	45	54	42	44	**55**	51	**58**	49	**56**	43	41	49	49.9	46	**52**	45	48	54	50.3	—	—

$\mathbf{2-20}D$	$\mathrm{DT_{EC}}$		$\mathrm{lq_{EC}}$		$\mathrm{lmm_{EC}}$	
#FEs/D	50	250	50	250	50	250
$\mathrm{DT_{EC}}$	—	—	50.4	**53**	**53**	**53**
$\mathrm{lq_{EC}}$	49.6	47	—	—	**52**	**50.2**
$\mathrm{lmm_{EC}}$	47	47	48	49.8	—	—

$\mathbf{2-20}D$	$\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathcal{M}}$	
#FEs/D	50	250	50	250	50	250	50	250
$\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	—	—	49	47	53	42	51	50.4
$\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	**51**	**53**	—	—	**54**	45	**52**	**53**
$\mathrm{lq}_{\mathcal{M}}$	47	**58**	46	**55**	—	—	48	**57**
$\mathrm{lmm}_{\mathcal{M}}$	49	49.6	48	47	52	43	—	—

was very successful on most of the functions, mainly on MULTI-MODAL WITH WEAK GLOBAL STRUCTURE and GRIEWANK-ROSENBROCK FUNCTION, whereas the $\mathrm{lmm_{EC}}$ with GP models was in general worse, though better on the functions f_6, STEP-ELLIPSOID f_7, and both SCHAFFERS $f_{17,18}$.

The EC providing the significantly better results than the other algorithms on the noiseless part of the COCO was the $\mathrm{lq_{EC}}$ especially in lower dimensions. On the contrary, the $\mathrm{lmm_{EC}}$ provided the overall worst performance, yielding better results only for the 250 FE/D budget on MULTI-MODAL FUNCTIONS f_{15-19} in 10 and $20D$. Also the $\mathrm{DT_{EC}}$ has shown very good results in higher dimensions.

Among the models, the $\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$ achieved the best result over all models. Moreover, both GP models have shown better results in higher dimensions than polynomial models, especially on ROSENBROCK FUNCTIONS $f_{8,9}$. An exception was only weak performance in the lower dimensions on the separable functions f_{1-5}. On the other hand, in the higher dimensions on separable functions, all models were outperformed by the overall worst $\mathrm{lq}_{\mathcal{M}}$. The $\mathrm{lmm}_{\mathcal{M}}$ scales with increasing dimension worse then the others, from being the best especially for 50 FE/D in $2D$ to being the worst results in $20D$.

Investigation on the Noisy COCO Benchmarks

Results on the noisy benchmarks from the COCO platform (Hansen et al., 2009a) are reported in Tables 37–60 and in Figures 37–60 in SUPPLEMENTARY MATERIAL. Their summary is in Table 25 and three examples in $10D$ are in Figure 37.

Table 26: A pairwise comparison of the evolution controls, models, and their combinations in $2D$, $4D$, $10D$, $12D$, $16D$ and $20D$ over the simulation benchmark for different evaluation budgets. The percentage of wins of i-th algorithm against j-th algorithm over all benchmark instances is given in the i-th row and j-th column. The numbers in bold mark the row algorithm being significantly better than the column algorithm according to the two-sided Wilcoxon signed rank test with the Holm correction at family-wise significance level $\alpha = 0.05$.

$\mathbf{2-20}D$	$\mathrm{DT}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{DT}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{DT}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{DT}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$		$\mathrm{lq}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{lq}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lq}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lq}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$	
#FEs/D	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250	50	250
$\mathrm{DT}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	—	—	50.3	52	**65**	**63**	47	46	**65**	58	64	57	63	41	57	54	37	41	36	40	50.3	39	49	36
$\mathrm{DT}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	49.7	48	—	—	**66**	62	52	40	**64**	58	**70**	56	**65**	39	60	54	43	35	42	40	50	36	50	37
$\mathrm{DT}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$	35	37	34	38	—	—	33	35	51	48	54	51	45	32	37	46	26	31	31	32	30	27	34	30
$\mathrm{DT}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$	53	54	48	**60**	**67**	**65**	—	—	**64**	**64**	**66**	**62**	**60**	50	55	61	38	47	41	47	51	37	49	44
$\mathrm{lq}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	35	42	36	42	49	52	36	36	—	—	52	44	46	36	37	44	28	32	34	32	37	28	35	31
$\mathrm{lq}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	36	43	30	44	46	49	34	38	48	56	—	—	42	43	40	51	28	41	34	39	38	39	40	40
$\mathrm{lq}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$	37	**59**	35	**61**	55	**68**	40	50	54	**64**	58	**57**	—	—	40	62	27	41	33	41	38	39	36	42
$\mathrm{lq}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$	43	46	40	46	**63**	54	45	39	**63**	56	60	49	60	38	—	—	36	37	37	38	43	32	42	39
$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	**63**	**59**	**57**	**65**	**74**	**69**	**62**	53	**72**	**68**	**72**	**59**	**73**	59	**64**	63	—	—	48	50	65	46	61	49.7
$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	**64**	**60**	58	**60**	**69**	**68**	59	53	**66**	**68**	**66**	**61**	**67**	59	63	**62**	52	50	—	—	64	42	56	51
$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$	49.7	**61**	50	**64**	**70**	**73**	49	**63**	**63**	**72**	**62**	**61**	**62**	**61**	57	**68**	35	54	36	58	—	—	44	54
$\mathrm{lmm}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$	51	**64**	50	**63**	**66**	**70**	51	**56**	**65**	**69**	**60**	**60**	**64**	58	**58**	**61**	39	50.3	44	49	56	46	—	—

$\mathbf{2-20}D$	$\mathrm{DT}_{\mathrm{EC}}$		$\mathrm{lq}_{\mathrm{EC}}$		$\mathrm{lmm}_{\mathrm{EC}}$	
#FEs/D	50	250	50	250	50	250
$\mathrm{DT}_{\mathrm{EC}}$	—	—	**58**	50.5	39	38
$\mathrm{lq}_{\mathrm{EC}}$	42	49.5	—	—	35	37
$\mathrm{lmm}_{\mathrm{EC}}$	**61**	**62**	**65**	**63**	—	—

$\mathbf{2-20}D$	$\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$		$\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$		$\mathrm{lq}_{\mathcal{M}}$		$\mathrm{lmm}_{\mathcal{M}}$	
#FEs/D	50	250	50	250	50	250	50	250
$\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$	—	—	50.1	49	**59**	49	49	47
$\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$	49.9	51	—	—	**59**	49.9	50.1	47
$\mathrm{lq}_{\mathcal{M}}$	41	51	41	50.1	—	—	39	50
$\mathrm{lmm}_{\mathcal{M}}$	51	**53**	49.9	**53**	**61**	50	—	—

The most successful combination on the noisy functions was $\mathrm{lq}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$ thanks to its adaptivity and perhaps also due to the more simple model. GP models combinations with the $\mathrm{DT}_{\mathrm{EC}}$ provide very good results especially in higher dimensions, unlike their weak combinations with the $\mathrm{lq}_{\mathrm{EC}}$, which has fast convergence only for the low budget on FUNCTIONS WITH MODERATE NOISE $f_{101-106}$. On the other hand, the $\mathrm{DT}_{\mathrm{EC}}+\mathrm{lq}_{\mathcal{M}}$ was dominant mainly while spending the full evaluation budget of 250 FE/D. The $\mathrm{lmm}_{\mathrm{EC}}+\mathrm{lmm}_{\mathcal{M}}$ has an overall weak performance except for very good results on DIFFERENT POWERS WITH GAUSSIAN NOISE f_{119}.

On FUNCTIONS WITH MODERATE NOISE $f_{101-106}$, the lower dimensions are dominated by the $\mathrm{lq}_{\mathrm{EC}}$, but it is gradually outperformed by the $\mathrm{DT}_{\mathrm{EC}}$ as the dimension grows. $\mathrm{lmm}_{\mathrm{EC}}$ overall weak performance especially when spending larger amounts of FEs is beaten only on FUNCTIONS WITH SEVERE NOISE $f_{107-121}$ in dimensions 10 and 20.

Among the models, very good results are achieved by the $\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$, whereas the performance of the $\mathrm{lmm}_{\mathcal{M}}$ is again decreasing as the dimension grows. Moreover, the fact that the performance of $\mathrm{GP}^{\mathrm{M}}_{\mathcal{M}}$ is significantly worse than that of $\mathrm{GP}^{\mathrm{PoI}}_{\mathcal{M}}$ indicates that taking into account uncertainty can be profitable on the noisy functions. Good result of the $\mathrm{lq}_{\mathcal{M}}$ at the end of the optimization process show that a simple model can deal with the noise more effectively than more complex ones if a sufficient budget is available.

Investigation on a Simulation Benchmark

The optimization of the simulation benchmark f_{sim} (Wu et al., 2016) was very time-consuming for all the tested combinations, especially in higher dimensions. Summarized results can be

seen in Table 26 and $\Delta_f^{\log}$ dependencies are depicted in $10D$ in Figure 37. The results in lower dimensions (Tables 61–63 and Figures 61–63 in SUPPLEMENTARY MATERIAL) show very few differences between the compared combinations. On the other hand, the higher dimensions are more convenient for lmm_{EC} (cf. Tables 64–67 and Figures 64–67 in SUPPLEMENTARY MATERIAL). Also the $\text{lmm}_{\mathcal{M}}$ has provided very good overall results not only in the original lmm-CMA version, but also in the $\text{DT}_{\text{EC}}+\text{lmm}_{\mathcal{M}}$ combination. This might suggest that the $\text{lmm}_{\mathcal{M}}$ is more convenient for real-world based simulation. On the other hand, the landscape of the tested benchmark can be easy to approximate for such a model.

4.8.3 *Conclusion*

We analysed three successful surrogate-assisted versions of the CMA-ES: the lmm-CMA, the DTS-CMA-ES, and the lq-CMA-ES, to separate the employed surrogate model and its evolution control. We implemented and assessed the performance of all the 12 combinations of the two quadratic, two GP models and three evolution controls on the noiseless and noisy parts of the COCO framework and on the buoy placement simulation benchmark in the expensive settings with a budget 250 FE/D.

We have found significant differences as to the performance of different evolution controls, models, and their combinations. The important finding was that both the model and the evolution control has significant influence on the convergence speed. The combinations originally designed together in the DTS-CMA-ES and lq-CMA-ES have provided the overall best results. However, mixing evolution controls of these two with other models shows that the evolution control itself may play the dominant role especially in the case of the lq-CMA-ES, the EC of which proved high performance with all the tested models. On the other hand, the linear-quadratic model mostly failed in combination with the remaining evolution controls, possibly because they were not designed for a global model. However, its simplicity played an important role in noisy cases where the linear-quadratic model ignores most of the noise if a sufficient budget is available.

The success of Gaussian process model using probability of improvement on noisy benchmarks indicates the role of surrogate models taking into consideration uncertainty. The performance of the otherwise worst lmm_{EC} on the simulation benchmark suggests that more experiments on real-world problems are needed.

4.9 COMBINING GAUSSIAN PROCESSES WITH NEURAL NETWORKS FOR THE CMA-ES

The importance of GPs in machine learning incited attempts to integrate them with the leading learning paradigm of the last decades – neural learning, including deep learning. The attractiveness of this research direction is further supported by recent theoretical results concerning relationships of asymptotic properties of important kinds of ANNs to properties of GPs (Lee et al., 2018; Matthews et al., 2018; Novak et al., 2019). The integration of GP with neural learning has been proposed on two different levels:

1. *Proper integration* of an ANN with a GP, in which the GP forms the final output layer of the ANN (Calandra et al., 2016; Wilson et al., 2016).

2. Only a *transfer of the layered structure*, which is a crucial feature, to the GP context, leading to the concept of deep GPs (Bui et al., 2016; Cutajar et al., 2017; Hebbal et al., 2018; Hernández-Muñoz et al., 2020).

The recalled research into the integration of GPs with neural learning has used regression data (Bui et al., 2016; Calandra et al., 2016; Cutajar et al., 2017; Hernández-Muñoz et al., 2020; Wilson et al., 2016), mostly from the UCI Machine Learning Repository (University of California, Irvine), but also data concerning locomotion of walking bipedal robots (Calandra et al., 2016), and face patches and hadwritten digits (Wilson et al., 2016). In (Cutajar et al., 2017; Hernández-Muñoz et al., 2020), also classification data were used. However, we are aware of only one application of such an integration, in particular of deep GPs, to two very easy 1- and 2-dimensional optimization problems (Hebbal et al., 2018). Hence, there is a gap between the importance of GPs in Bayesian optimization and missing investigations of the suitability of integrated GP-ANN models for surrogate modeling in black-box optimization. That gap motivated the research reported in this section summarizing our papers concerning this topic (Koza et al., 2021a,b; Růžička et al., 2021).

We have performed this novel kind of investigation of GP-ANN (see paragraph GP as the Output Layer of ANN in Section 2.3.1) integration to the DTS-CMA-ES on the noiseless part of the COCO framework (Hansen et al., 2009b, 2021).

4.9.1 *Using Past Experience for GP Configuration*

The covariance function of a GP is crucial for modeling relationships between observations corresponding to different points in the input space. This problem consists in the fact that although the quality of the GP depends solely on its predictions and the true values of the black-box objective function, to obtain the prediction needs in addition a complete run of an optimizer, which is typically much more demanding. The problem is particularly serious when combing GPs and ANNs because then separate runs are needed for all considered combinations of GP covariance functions with the considered ANN topologies. That motivated the research we reported in (Koza et al., 2021a,b), in which we have investigated 10 GP configurations, differing primarily through the choice of the covariance function: Instead of running the optimizer specifically for the configuration task, we used comprehensive data from previous runs of the DTS-CMA-ES on the noiseless part of the COCO framework (Hansen et al., 2009b, 2021).

Employed Data from DTS-CMA-ES Runs

As a dataset to compare different configurations of the combined GP-ANN model, we used recorded DTS-CMA-ES runs from previous experiments. This allowed us to effectively evaluate the surrogate model on its own without having to perform the whole optimization. Our open-source Matlab implementation of DTS-CMA-ES, used to obtain the data, is available at (Bajer and Pitra, 2014). We have utilized Gaussian processes with 8 different covariance functions (κ_{LIN}, κ_{Q}, κ_{RQ}, κ_{SE}, $\kappa_{\text{Mat}}^{5/2}$, κ_{NN}, κ_{Gibbs}, and composite $\kappa_{\text{SE+Q}}$) implemented using the GPML Toolbox (Rasmussen and Nickisch). The optimization runs were collected on the noiseless part of the COCO platform (Hansen et al., 2009b, 2021) in dimensions 2, 3, 5, 10, and 20 in 5 different instances. Therefore, for each combination of benchmark function and dimension, 40 runs are available. More important than the number of runs, however, is the number of their

generations because data from each generation apart from the first can be used for testing all those surrogate models that could be trained with data from the previous generations. The number of generations in a particular run of DTS-CMA-ES algorithm is unknown before the run and depends on the objective function and its particular instance. We have omitted the Linear slope function f_5, which is easy to optimize, and the recorded runs did not provide enough data samples to train and evaluate the surrogate models. This resulted in total number of 4600 available runs. The benchmarks and their total numbers of available generations for individual dimensions are listed in Table 27.

Due to the way how the DTS-CMA-ES works (see Section 4.4), the collected sequences have several specific properties:

1. The first GP $(\mathcal{M}_1(\mathbf{x}))_{\mathbf{x}\in\mathcal{X}}$ trained after the g-th generation of the CMA-ES has as training data only pairs $(\mathbf{x}, \mathrm{bb}(\mathbf{x}))$ in which the true value $\mathrm{bb}(\mathbf{x})$ was obtained before the g-th generation. It does not depend on the results of the CMA-ES in the g-th and later generations.

2. The second GP $(\mathcal{M}_2(\mathbf{x}))_{\mathbf{x}\in\mathcal{X}}$ trained after the g-th generation of the CMA-ES has as training data pairs $(\mathbf{x}, \mathrm{bb}(\mathbf{x}))$ in which the true value $\mathrm{bb}(\mathbf{x})$ was obtained before the g-th generation, as well as the pairs $(\mathbf{a}, \mathrm{bb}(\mathbf{a}))$ in which $\mathbf{a}$ is one of the points $\mathbf{a}_1, \ldots, \mathbf{a}_{\lceil\alpha(g)\lambda\rceil}$ selected with active learning among the points $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ generated in the g-th generation. It does not depend on the results of the CMA-ES in the $g+1$-st and later generations.

3. The CMA-ES in the $g+1$-st generation depends on the values $\mathrm{bb}(\mathbf{a})$ in the points $\mathbf{a} \in \{\mathbf{a}_1, \ldots, \mathbf{a}_{\lceil\alpha(g)\lambda\rceil}\}$ selected with active learning and on the values $\mathcal{M}_2(\mathbf{x})$ for $\mathbf{x} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_\lambda\} \setminus \{\mathbf{a}_1, \ldots, \mathbf{a}_{\lceil\alpha(g)\lambda\rceil}\}$. Due to 2., it indirectly depends also on $(\mathbf{x}, \mathrm{bb}(\mathbf{x}))$ with $\mathbf{x}$ generated in an earlier than g-th generation.

Experimental Setup of Empirical Investigation of GP Configurations

In the experiments described below, we used two different libraries implementing Gaussian processes. For those without an ANN, we used Matlab's toolbox GPML (Rasmussen and Nickisch) that accompanies the book by Rasmussen and Williams (2006). Except the spectral mixture kernel κ_{SM}, which was implemented in Python using GPyTorch library as well as the combination of GP and ANN (Gardner et al., 2019).

We have compared six surrogate models combining neural networks with Gaussian processes using the following covariances: κ_{LIN}, κ_{Q}, κ_{RQ}, κ_{SE}, $\kappa_{\mathrm{Mat}}^{5/2}$, and $\kappa_{\mathrm{SE+Q}}$. Then we have compared the winning ANN-GP combination and Gaussian processes with nine different covariance functions: κ_{LIN}, κ_{Q}, κ_{RQ}, κ_{SE}, $\kappa_{\mathrm{Mat}}^{5/2}$, κ_{NN}, κ_{Gibbs}, $\kappa_{\mathrm{SE+Q}}$, and κ_{SM}. As to the combination ANN-GP, we decided to train it on the same set of training data $\mathcal{T}$ in step 2 of Algorithm 5 using TSS nearest in area of maximal range $r_{\max}$ as was used in steps 4 and 9 of Algorithm 9 (Bajer et al., 2019). Due to the $r_{\max}$ limitation, this set is rather restricted and allows training only a rather restricted ANN. Therefore, we decided to use a multilayer perceptron with a single hidden layer, thus a topology (n_I, n_H, n_O), where n_I is the dimension of the training data, i. e., $n_I \in \{2, 3, 5, 10, 20\}$, and

$$n_H = n_O = \begin{cases} 2 & \text{if } n_I = 2, \\ 3 & \text{if } n_I = 3, 5, \\ 5 & \text{if } n_I = 10, 20. \end{cases} \tag{102}$$

Table 27: Noiseless benchmark functions of the COCO platform and the number of available generations of DTS-CMA-ES runs for each of them in each considered dimension.

	COCO function	Number of available generations				
		2D	3D	5D	10D	20D
	Separable					
f_1	Sphere	4689	6910	11463	17385	25296
f_2	Separable Ellipsoid	6609	9613	15171	25994	55714
f_3	Separable Rastrigin	7688	11308	17382	27482	42660
f_4	Büche-Rastrigin	8855	13447	22203	31483	49673
	Moderately Ill-Conditioned					
f_6	Attractive Sector	16577	25200	38150	45119	72795
f_7	Step Ellipsoid	7103	9816	24112	34090	56340
f_8	Rosenbrock	7306	11916	21191	32730	71754
f_9	Rotated Rosenbrock	7687	12716	24084	35299	71017
	Highly Ill-Conditioned					
f_{10}	Ellipsoid with High Conditioning	6691	9548	15867	25327	59469
f_{11}	Discus	6999	9657	15877	25478	45181
f_{12}	Bent Cigar	10369	18059	28651	34605	56528
f_{13}	Sharp Ridge	7760	11129	20346	30581	48154
f_{14}	Different Powers	6653	10273	17693	31590	61960
	Multi-modal with Global Structure					
f_{15}	Non-separable Rastrigin	7855	11476	19374	28986	44446
f_{16}	Weierstrass	9294	13617	24158	27628	40969
f_{17}	Schaffers F7	9031	13960	24244	34514	56247
f_{18}	Ill-Conditioned Schaffers F7	9598	13404	25802	31609	53836
f_{19}	Composite Griewank-Rosenbrock	9147	16268	24456	34171	53536
	Multi-modal Weakly Structured					
f_{20}	Schwefel	9081	13676	24219	33753	53104
f_{21}	Gallagher's Gaussian 101-me Points	7645	12199	18208	25366	43186
f_{22}	Gallagher's Gaussian 21-hi Points	7629	11086	17881	26626	44971
f_{23}	Katsuura	8751	11233	17435	25030	37366
f_{24}	Lunacek bi-Rastrigin	8983	13966	19405	29762	44420

Table 28: Statistical comparison of six ANN-GP models using different covariance functions accross data from noiseless COCO functions. It shows for how many functions (23 in total) was the covariance in a row significantly better than the one in a column.

	κ_{LIN}	κ_{Q}	κ_{RQ}	κ_{SE}	$\kappa_{\text{Mat}}^{5/2}$	$\kappa_{\text{SE+Q}}$	Σ
κ_{LIN}	–	20	19	21	20	20	100
κ_{Q}	0	–	3	2	3	16	24
κ_{RQ}	1	5	–	1	1	13	21
κ_{SE}	2	6	1	–	1	14	24
$\kappa_{\text{Mat}}^{5/2}$	2	6	2	0	–	13	23
$\kappa_{\text{SE+Q}}$	2	1	2	0	1	–	6

Table 29: Statistical comparison of six ANN-GP models using different covariance functions accross data from noiseless COCO functions. It shows for how many combinations of input dimension and a specific function group (25 in total) was the covariance in a row significantly better than the one in a column.

	κ_{LIN}	κ_{Q}	κ_{RQ}	κ_{SE}	$\kappa_{\text{Mat}}^{5/2}$	$\kappa_{\text{SE+Q}}$	Σ
κ_{LIN}	–	25	23	21	20	24	113
κ_{Q}	0	–	4	0	3	16	23
κ_{RQ}	0	4	–	2	1	13	20
κ_{SE}	0	4	5	–	5	13	27
$\kappa_{\text{Mat}}^{5/2}$	0	2	3	1	–	9	15
$\kappa_{\text{SE+Q}}$	0	0	0	0	0	–	0

As the activation function for both the hidden and output layer, we chose a logistic sigmoid.

We trained the weights and biases of the neural network together with the parameters of the Gaussian process as proposed by Wilson et al. (2016) and outlined in paragraph GP as the Output Layer of ANN in Section 2.3.1. As a loss function, we used the Gaussian log-likelihood and optimized the parameters with Adam (Kingma and Ba, 2015) for a maximum of 1000 iterations. We also kept a 10% validation set out of the training data to monitor overfitting, and we selected the model with the lowest L_2 validation error during the training.

Results of GP Configuration Investigation

We evaluated six different models on every generation of samples listed in Table 27. The utilized metric was err_{RDE}, which shows how precise the model is in ordering of the predicted values. A Non-parametric Friedman test was conducted on the values of err_{RDE} across all results for particular functions and function types, and for a particular combination of dimensions and function types. If the null hypothesis of the equality of all six considered methods was declined, the Wilcoxon signed-rank test was performed for all pairs of covariance functions, and its results were corrected for multiple hypotheses testing using the Holm method. We summarized the results of statistical testing on six ANN-GP combinations in Tables 28 and 29. The results show that the combined model performs best with the simplest linear kernel and therefore, only the κ_{LIN} is used in the following comparisons.

The results of comparison between 9 GP models and 1 ANN-GP model are presented first in a function-method view in Table 30, then in a dimension-method view in Table 31. The

interpretation of the results in both tables is the same. Similarly to previous comparison, a non-parametric Friedman test was conducted on values of $\mathrm{err}_{\mathrm{RDE}}$ across all results for particular functions and function types in Table 30, and for a particular combination of dimensions and function types in Table 31. If the null hypothesis of the equality of all ten considered methods was declined, a post hoc test was performed, and its results were corrected for multiple hypotheses testing using the Holm method. In the tables, we highlight bold those methods that are not significantly different from the method achieving the lowest $\mathrm{err}_{\mathrm{RDE}}$. Additionally, the number $n \in \mathbb{N}$ of other methods that achieved a significantly worse $\mathrm{err}_{\mathrm{RDE}}$ is reported using the ^{*n} notation.

If we look closely at the results in Table 30, we can see that the lowest values of the average $\mathrm{err}_{\mathrm{RDE}}$ are achieved using the κ_{RQ} in most cases. Specifically, the rational quadratic covariance κ_{RQ} performs best on 16 functions, Matérn $5/2$ $\kappa_{\mathrm{Mat}}^{5/2}$ on 3, squared exponential κ_{SE} on 2, and quadratic κ_{Q} on 2 functions out of the total number of 23. Interestingly, the two functions on which the κ_{Q} is the best performing are Attractive sector functionf_6 and Step ellipsoid f_7, which are both based on quadratic functions Finck et al. (2009). This suggests that the quadratic kernel can indeed adequately capture the functions with the global structure of quadratic polynomial. On the other hand, the $\kappa_{\mathrm{Mat}}^{5/2}$ is the best performing on functions f_{20}, f_{23}, and f_{24}, which are highly multimodal and therefore hard to optimize.

Similar results can also be seen in Table 31, where the values of $\mathrm{err}_{\mathrm{RDE}}$ are grouped by the input space dimension. The κ_{RQ} is the best in the 18 cases, $\kappa_{\mathrm{Mat}}^{5/2}$ in 4, κ_{Q} in 2, and κ_{SE} in 1 case. Again κ_{SE} was the best when solving highly multimodal weakly structured functions. We also analyzed the correlation between the dimensionality of the problem and the resulting average $\mathrm{err}_{\mathrm{RDE}}$ for every considered method. For this purpose, we computed Pearson's and Spearman's correlation coefficients. Both coefficients detected a statistically significant positive correlation in the linear, quadratic, Gibbs, and spectral mixture covariances. The correlation of κ_{SE} was found significant only by the Pearson's coefficient. Interestingly, both methods discovered a significant negative correlation for the neural network covariance function κ_{NN}.

However, it is important to note here that the best three covariances: κ_{RQ}, κ_{SE}, and $\kappa_{\mathrm{Mat}}^{5/2}$ are statistically equivalent in all cases according to the Friedman-posthoc test with Holm correction.

Finally, the combination ANN-GP with the linear covariance was statistically equivalent to the best-performing covariance on six different functions. In those cases, it performed better than just GP with the linear kernel. However, in the rest of the benchmark functions, the linear covariance function produced better results without the ANN.

4.9.2 *Conclusion*

We found out that if we compare the combined ANN-GP with GP both with linear covariance function, the neural extensions can bring better results in some cases. However, using more complex covariance functions with GP is still better. Unfortunately, the results with other covariances in combination with ANN ended up much worse.

The results in Tables 30 and 31 reveal that the lowest $\mathrm{err}_{\mathrm{RDE}}$ values are most often achieved using the covariance functions rational quadratic, squared exponential, and Matérn $5/2$. Moreover, these three covariance functions are always equivalent, in the sense that among the $\mathrm{err}_{\mathrm{RDE}}$ values achieved with them, no pair has been found significantly different by the performed Friedman-posthoc test with Holm correction for simultaneous hypotheses testing. Occasion-

Table 30: Comparison of average err_{RDE} values for different surrogate models depending on a particular benchmark function. There are nine different covariance functions for Gaussian processes and the ANN-GP combination with a linear kernel. Those methods that were not significantly different from the best performing are marked in bold. The number in the upper index indicates the number of methods that performed significantly worse.

		GP									ANN
function		κ_{LIN}	κ_{Q}	κ_{RQ}	κ_{SE}	$\kappa_{\text{Mat}}^{5/2}$	κ_{NN}	κ_{Gibbs}	$\kappa_{\text{SE+Q}}$	κ_{SM}	κ_{LIN}
Separable Functions	f_1	.238	.193	**.056**^{*6}	**.063**^{*6}	**.070**^{*4}	.168	**.130**^{*1}	.161	.225	.242
	f_2	.231	.181	**.096**^{*5}	**.098**^{*5}	**.099**^{*5}	.187	**.153**^{*2}	.222	.280	**.128**^{*2}
	f_3	.236	**.185**^{*1}	**.142**^{*5}	**.147**^{*4}	**.160**^{*4}	**.185**^{*1}	.202^{*1}	.224	.503	.300
	f_4	.251	**.170**^{*2}	**.134**^{*6}	**.160**^{*3}	**.153**^{*4}	.189^{*1}	.213^{*1}	.224	.471	.352
	all	.239	.182^{*2}	**.107**^{*7}	**.117**^{*7}	**.121**^{*7}	.182^{*1}	.175^{*4}	.208	.370	.256
Moderate conditioning	f_6	**.209**^{*2}	**.138**^{*5}	**.203**^{*3}	.215^{*2}	**.208**^{*2}	**.199**^{*3}	.246	.254	.440	.351
	f_7	**.231**^{*1}	**.185**^{*3}	**.194**^{*2}	**.224**^{*1}	**.206**^{*1}	**.192**^{*3}	.249	.243	.438	**.200**^{*2}
	f_8	.231	**.183**	**.129**^{*4}	**.138**^{*3}	**.143**^{*2}	.182	**.184**^{*1}	.198	**.187**^{*1}	.307
	f_9	.206	**.179**	**.128**^{*4}	**.140**^{*3}	**.140**^{*3}	.175	**.170**^{*3}	**.167**^{*1}	.263	.307
	all	.219^{*1}	**.171**^{*4}	**.163**^{*5}	**.179**^{*4}	**.174**^{*4}	**.187**^{*3}	.212^{*2}	.216	.342	.291
High conditioning and unimodal	f_{10}	.235	.179	**.095**^{*5}	**.101**^{*5}	**.099**^{*5}	.182	**.141**^{*1}	.209	.295	**.142**^{*1}
	f_{11}	.243	**.152**^{*1}	**.105**^{*4}	**.114**^{*4}	**.115**^{*4}	.196	**.144**^{*2}	.241	.357	**.147**^{*2}
	f_{12}	**.183**	**.166**^{*2}	**.145**^{*4}	**.153**^{*3}	**.151**^{*3}	.196	**.187**^{*2}	.236	.290	.257
	f_{13}	.231	.202	**.116**^{*4}	**.208**^{*1}	**.220**^{*1}	**.169**^{*1}	**.177**^{*1}	.210	.375	**.198**^{*1}
	f_{14}	.234	.211	**.140**^{*5}	**.165**^{*4}	**.162**^{*4}	**.187**^{*2}	**.204**^{*2}	.227	.310	.289
	all	.225^{*1}	.182^{*2}	**.120**^{*7}	**.148**^{*6}	**.149**^{*6}	.186^{*2}	.171^{*5}	.224	.326	.205^{*2}
Multi-modal adequate global structure	f_{15}	.242	**.172**^{*3}	**.144**^{*4}	**.141**^{*4}	**.173**^{*2}	**.182**^{*2}	**.200**^{*1}	.215	.488	.316
	f_{16}	**.221**^{*1}	**.189**^{*2}	**.172**^{*3}	**.169**^{*3}	**.173**^{*3}	**.185**^{*2}	**.207**^{*2}	.234	.492	.551
	f_{17}	.242	**.199**^{*2}	**.160**^{*5}	**.211**^{*2}	**.185**^{*2}	**.189**^{*2}	.227^{*1}	.245	.463	.391
	f_{18}	.259	**.203**^{*2}	**.157**^{*4}	**.207**^{*2}	**.182**^{*4}	**.191**^{*2}	**.210**^{*2}	.245	.480	.380
	f_{19}	.232^{*1}	**.206**^{*2}	**.143**^{*5}	**.154**^{*4}	**.148**^{*4}	**.200**^{*2}	.249	.217^{*1}	.537	.397
	all	.239^{*2}	.194^{*4}	**.155**^{*7}	**.177**^{*5}	**.172**^{*5}	.189^{*4}	.218^{*2}	.231^{*2}	.493	.408
Multi-modal weak global structure	f_{20}	.237	**.164**^{*3}	**.161**^{*2}	**.174**^{*1}	**.155**^{*4}	.191	**.218**	.216	.383	**.210**^{*1}
	f_{21}	.214	**.197**^{*1}	**.145**^{*4}	**.170**^{*2}	**.152**^{*4}	**.184**^{*2}	**.169**^{*2}	.206	.372	.389
	f_{22}	.203	.215	**.128**^{*6}	**.155**^{*3}	**.145**^{*5}	.192	**.160**^{*2}	.224	.391	.347
	f_{23}	.237	**.208**^{*1}	**.168**^{*3}	**.156**^{*3}	**.147**^{*3}	**.213**^{*1}	**.210**^{*1}	**.206**^{*1}	.543	.589
	f_{24}	.229	**.209**^{*1}	**.144**^{*3}	**.127**^{*4}	**.118**^{*4}	**.197**^{*1}	.217	**.220**	.545	.446
	all	.224^{*2}	.199^{*2}	**.149**^{*7}	**.156**^{*7}	**.144**^{*7}	.195^{*2}	.195^{*2}	.214^{*2}	.442	.397

Table 31: Comparison of average err_{RDE} values for different surrogate models depending on the type of benchmark function and the input space dimension. There are nine different covariance functions for Gaussian processes and the ANN-GP combination with a linear kernel. Those methods that were not significantly different from the best performing are marked in bold. The number in the upper index indicates the number of methods that performed significantly worse.

	function	GP									ANN
	group	κ_{LIN}	κ_{Q}	κ_{RQ}	κ_{SE}	$\kappa_{\text{Mat}}^{5/2}$	κ_{NN}	κ_{Gibbs}	$\kappa_{\text{SE+Q}}$	κ_{SM}	κ_{LIN}
$2D$	SEP	.172	**.162**	**.097**^{*5}	**.106**^{*4}	**.110**^{*4}	.179	**.135**^{*1}	.198	.280	.252
	MOD	**.160**^{*2}	**.154**^{*2}	**.174**^{*2}	**.167**^{*2}	**.201**	**.189**	**.182**^{*1}	**.194**	.286	.341
	HC	**.157**^{*2}	**.177**	**.129**^{*4}	**.135**^{*4}	**.147**^{*3}	.212	**.142**^{*4}	.222	.227	.204
	MMA	**.187**^{*2}	**.179**^{*2}	**.155**^{*3}	**.170**^{*2}	**.179**^{*2}	**.200**^{*2}	**.195**^{*2}	.231	.437	.416
	MMW	**.184**^{*2}	**.182**^{*2}	**.137**^{*3}	**.140**^{*4}	**.148**^{*3}	**.205**	**.169**^{*2}	.221	.414	.443
	all	.172^{*3}	.172^{*3}	**.139**^{*6}	**.144**^{*6}	**.157**^{*4}	.198^{*2}	**.165**^{*4}	.215^{*1}	.333	.334
$3D$	SEP	.204	**.175**^{*1}	**.114**^{*5}	**.117**^{*5}	**.123**^{*5}	.196	**.163**^{*1}	.218	.378	.258
	MOD	**.201**	**.160**^{*3}	**.172**^{*3}	**.177**^{*3}	**.173**^{*3}	**.198**	**.187**^{*1}	.226	.342	.313
	HC	.210	**.173**^{*1}	**.132**^{*4}	**.141**^{*3}	**.143**^{*3}	.199	**.149**^{*3}	.234	.264	**.174**^{*1}
	MMA	.227	**.198**^{*2}	**.159**^{*4}	**.164**^{*4}	**.180**^{*3}	**.193**^{*2}	**.201**^{*2}	.232	.458	.353
	MMW	.215	**.197**^{*1}	**.148**^{*4}	**.155**^{*2}	**.145**^{*5}	.208	**.204**^{*1}	.213^{*1}	.427	.387
	all	.214^{*1}	.182^{*4}	**.145**^{*7}	**.153**^{*6}	**.154**^{*6}	.199^{*1}	.180^{*4}	.223^{*1}	.360	.278^{*1}
$5D$	SEP	.241	.189^{*1}	**.111**^{*7}	**.118**^{*5}	**.121**^{*5}	.193	.171^{*1}	.226	.421	.270
	MOD	.249	**.174**^{*3}	**.157**^{*5}	**.168**^{*3}	**.159**^{*4}	**.195**^{*1}	.205	.224	.387	.297
	HC	.241	.180^{*1}	**.119**^{*6}	**.136**^{*3}	**.136**^{*3}	.185^{*1}	**.148**^{*3}	.240	.337	.202^{*1}
	MMA	.251	**.189**^{*2}	**.149**^{*5}	**.167**^{*4}	**.169**^{*4}	**.183**^{*2}	.206^{*2}	.240	.478	.373
	MMW	.241	.205^{*1}	**.140**^{*4}	**.143**^{*4}	**.136**^{*6}	.202^{*1}	**.185**^{*2}	.209	.459	.363
	all	.247^{*1}	.185^{*4}	**.136**^{*7}	**.148**^{*7}	**.146**^{*7}	.190^{*4}	.177^{*4}	.229^{*1}	.403	.289^{*1}
$10D$	SEP	.271	.203^{*1}	**.123**^{*4}	**.117**^{*5}	**.127**^{*4}	**.188**^{*1}	**.183**^{*1}	.216	.456	.293
	MOD	.256	**.195**^{*2}	**.151**^{*5}	**.167**^{*4}	**.155**^{*5}	**.192**^{*2}	.243	.228	.407	.298
	HC	.264	.198^{*1}	**.111**^{*5}	**.159**^{*1}	**.156**^{*2}	**.179**^{*1}	**.166**^{*1}	.230	.412	.206^{*1}
	MMA	.274	**.204**^{*2}	**.159**^{*5}	**.175**^{*4}	**.162**^{*4}	**.187**^{*3}	.224^{*1}	.233	.538	.422
	MMW	.246	.206	**.139**^{*5}	**.146**^{*4}	**.131**^{*6}	**.183**^{*2}	.211	.215	.454	.363
	all	.264^{*1}	.200^{*3}	**.138**^{*7}	**.155**^{*6}	**.149**^{*6}	.185^{*4}	.205^{*3}	.226^{*1}	.443	.313^{*1}
$20D$	SEP	.288	**.196**	**.109**^{*4}	**.132**^{*4}	**.120**^{*4}	**.168**^{*3}	.250	**.188**	.445	.348
	MOD	**.228**	**.197**^{*1}	**.143**^{*2}	**.173**	**.145**^{*2}	**.174**^{*2}	**.245**	**.206**	.523	.353
	HC	**.253**	**.184**	**.111**^{*1}	**.171**	**.164**^{*1}	**.155**	**.249**	**.197**	.471	**.255**
	MMA	.258	**.201**^{*2}	**.155**^{*4}	**.206**^{*2}	**.170**^{*2}	**.184**^{*2}	.266	**.220**	.596	.480
	MMW	**.234**	**.204**	**.181**	**.198**	**.160**	**.179**	**.205**	**.213**	**.538**	**.436**
	all	.249	.194^{*3}	**.140**^{*6}	**.180**^{*4}	**.156**^{*5}	**.169**^{*4}	.244	.203^{*2}	.513	.375

ally, they are also equivalent to other configurations, e. g., to the covariance functions linear or quadratic, or even to the combination ANN-GP. In addition, we have also shown in (Růžička et al., 2021), statistical equivalency of rational quadratic, squared exponential, and Matérn $5/2$, in the sense that for no function, no group of functions, no dimension, and no function-dimension combination, none of these covariance functions was significantly better than the other in ANN-GP surrogate model for the DTS-CMA-ES running on the entire noiseless part of the COCO platform. To our knowledge, the comparison of GP covariance functions presented in (Koza et al., 2021b) is the first in the context of optimization data. Therefore, it is not surprising that the conclusions drawn from it are not identical to conclusions drawn from comparisons performed with other kinds of data (Calandra et al., 2016; Duvenaud, 2014; Kronberger and Kommenda, 2013; Wilson and Adams, 2013; Wilson et al., 2016).

5 CONTRIBUTIONS TO LANDSCAPE ANALYSIS OF SURROGATE MODELS

This chapter partially cites the text of the articles (Pitra et al., 2018b) and (Pitra et al., 2022).

Our research of surrogate modeling in evolutionary black-box optimization context has shown that no surrogate model or algorithm using the model improved the CMA-ES significantly better than all other proposed surrogate model approaches in the benchmark expensive scenario (Bajer et al., 2019; Pitra et al., 2017c). This finding was furthermore supported in (Pitra et al., 2021), where we have shown that different surrogate models, their controls, and their combinations can show significant differences in the performance of the same optimizer on different data. Moreover, the predictive accuracy of individual models is strongly influenced by the choice of their parameters. Therefore, in this chapter we investigate the relationships between the settings of individual models and the optimized data for better understanding of the whole surrogate modeling task.

In Section 5.1, we study the connection between two surrogate models in 29 different settings and features describing fitness landscapes (see Section 2.5). We provide the research in static settings, published in (Pitra et al., 2018b), where the perfomance of surrogate models on fitness is measured on the fixed number of function evaluations sampled for each problem instance (Saini et al., 2019). In Section 5.2, we present the investigation of landscape feature properties in connection with surrogate models on the data from the actual runs of the surrogate-assisted version of the CMA-ES, published in (Pitra et al., 2022). The properties of the utilized data are completely different from the sampled data in the previous section implying different values of landscape features as emphasized in (Renau et al., 2020). In final Section 5.3, we investigate the relationships of the landscape features to four kinds of surrogate models in different settings using three TSS methods for training data selection in the same context as the previous section. The investigation was published in (Pitra et al., 2022).

5.1 RELATIONSHIPS OF SURROGATE MODELS TO FITNESS LANDSCAPES IN EXPENSIVE OPTIMIZATION

In this section, we present the study, published in (Pitra et al., 2018b), investigating the relations between the two surrogate models capable to predict the whole distribution of function values – Gaussian processes (Rasmussen and Williams, 2006) and random forests (Breiman, 2001) – and the properties of fitness landscapes. Considering that the motivation of such investigation consists in saving function evaluations during the run of the optimization algorithm on the expensive continuous black-box function, we construct the models using the budget $50D$ evaluations of all 24 functions from the noiseless part of COCO benchmark set (Hansen et al., 2009b). Thus, the research is conducted in static settings, where the performance of a surrogate model is measured in connection with the entire landscape of the fitness function, i. e., similarly to (Saini et al., 2019). Furthermore, due the to a limited amount of function evaluations, we use 11 sets of landscape feature that do not require additional function evaluations

(unlike e. g., the features related to local searches conducted from sampled points (Mersmann et al., 2011)) and don't have low reliability in high dimensions, where the input space is sparse (unlike e. g., Generalized Cell-Mapping features (Kerschke et al., 2014)). We perform a statistical analysis of the relationships between the predictive performance of various settings of the considered surogate models and the landscape features calculated on individual benchmarks.

5.1.1 *Experimental Investigation of Landscape Analysis for random forests and Gaussian processes*

In this section, the performance of RF and GP surrogate models with various settings is investigated in the context of fitness landscape features of the datasets generated from sampling on noiseless benchmark functions.

Data

The considered experiments were performed[1] on the set of all 24 noiseless functions from the COCO framework (Hansen et al., 2009b) in dimensions $D = \{2, 5, 10\}$ using random sampling from 15 different instances per function. The datasets consisting of $50D$ points for each instance per function were generated by a random improved Latin Hypercube design (Beachkofski and Grandhi, 2002) covering the input space $[-5, 5]^D$. Such sample of evaluations has been shown sufficient for certain high-level properties by means of numerical features by Kerschke et al. (2016), which can be convenient especially in lower dimensions, where the initial phase of optimization run takes part within such budget of expensive evaluations. The overall predictive performance of the surrogate models was tested through 5-fold cross-validation on the generated datasets.

Experimental Setup

GAUSSIAN PROCESS MODEL The GP regression model was employed using constant mean $\mu(\mathbf{x}) = \text{mean}(\mathbf{y})$, where $\mathbf{y}$ are the training values, and 7 different isotropic covariance functions: κ_{RQ}, κ_{SE}, $\kappa_{\text{Mat}}^{1/2}$, $\kappa_{\text{Mat}}^{3/2}$, $\kappa_{\text{Mat}}^{5/2}$, κ_{NN}, and κ_{ADD}. κ_{RQ} and κ_{SE} were also used in ARD versions ($\kappa_{\text{RQ}}^{\text{ARD}}$ and $\kappa_{\text{SE}}^{\text{ARD}}$). The hyperparameters were optimized with MATLAB's `fmincon` using 5 optimization trials, except for the additive covariance function κ_{ADD}, which was optimized with only 3 trials due to its relatively high complexity.

The neural network covariance (31) was specified with isotropic distance measure $\mathbf{P} = \ell^{-2}\mathbf{I}$. The additive kernel (37) used one-dimensional squared exponential (26) as the basis.

The rest of initial values for hyperparameters, together with their bounds are reported Table 32. The initial values for repeated optimization trials were sampled by Latin Hypercube design.

RANDOM FOREST MODEL The RF models were tested using the full-factorial desing on the ensemble method, splitting method and error gain functions err of the decision tree. In addition, the number of trees n_{tree}, the number of points N_t, and the number of dimensions used for training the individual tree n_D were sampled from the values in Table 33. Thus, the RF experimental part sampled RF models from 1600 different settings. Bagging (Breiman, 1996) and boosting (Chen and Guestrin, 2016) were tested as ensemble methods (see paragraphs Bagging and Boosting in Section 2.3.1).

1 the source code is freely available at https://github.com/bajeluk/surrogate-cmaes/tree/meta

Table 32: Experimental settings of GP. σ_n and ℓ apply to all covariances. σ_f applies to all except the κ_{ADD}, in which $\sigma_f^{(r)}$ scales each degree $r \in R$, $R = \{1,2,3,5,7,10\} \cap \{1,\ldots,D\}$ of interaction separately. $\sigma_f^{(r)}$ and its upper bound were initialized proportionally to $\binom{D}{r}$.

GP hyperparam	initial value	constrains
σ_n	1e−2	$[1e{-}3, 1e1]$
ℓ	$\text{std}(\mathbf{X})$	$[1e{-}2, 1e2]$
σ_f	$\frac{\text{std}(\mathbf{y})}{\sqrt{2}}$	$[1e{-}2, 1e6]$

Table 33: Experimental settings of RF: n_{tree} – number of trees in RF, N_t, n_D – number of points and dimensions for training individual tree, N – number of available training points, D – input space dimension. Split methods and error gain functions err are tested using full-factorial design, n_{tree}, N_t, and n_D are sampled.

RF parameter	bagging	boosting
err	$\{\text{err}_{\text{MSE}}, \text{err}_{\sigma^2}, \text{err}_{\text{NN}}\}$	err_{Grad}
split	{CART, SECRET, OC1, SUPPORT, PAIR}	
n_{tree}	$\{64, 128, 256, 512, 1024\}$	
N_t	$\lceil \{0.25, 0.5, 0.75, 1\} \cdot N \rceil$	
n_D	$\lceil \{0.25, 0.5, 0.75, 1\} \cdot D \rceil$	

Decision trees in bagged RF were employed with the following error gain functions MSE err_{MSE}, variance of predicted y-values err_{σ^2}, and nearest-neighbor entropy estimator err_{NN} were employed as error gain functions (see Equations (48), (49), and (51) respectively). In bagged RF, cross-validation pruning (Breiman, 1984) was utilized to optimize the tree structure. In addition, the following five regression models were used in leaves: constant, linear, linear with interactions, pure quadratic (quadratic without interactions), and full quadratic. The model providing the best fit according to the MSE loss function was always selected for the relevant leaf and appropriate data. In boosted RF, the maximum tree depth was set to 8, in accordance with (Chen and Guestrin, 2016).

Considering decision tree settings regardless the ensemble method, the five splitting methods from the following algoritms were employed (see paragraph Split methods in Section 2.3.1): CART (Breiman, 1984), SECRET (Dobra and Gehrke, 2002), OC1 (Murthy et al., 1994), SUPPORT (Chaudhuri et al., 1994), and a method from (Hinton and Revow, 1996) (PAIR). The remaining decision tree parameters have been taken identical to settings from (Pitra et al., 2018a).

The 91 calculated landscape features were from the following 11 feature sets (Kerschke, 2017b) (see definitions in Appendix A):

Basic	Φ_{Basic},	*Levelset*	Φ_{Lvl},
CM Angle	$\Phi_{\text{CM-Angl}}$,	*Meta-Model*	Φ_{MM},
CM Convexity	$\Phi_{\text{CM-Conv}}$,	*Nearest Better Clustering*	Φ_{NBC},
CM Gradient Homogeneity	$\Phi_{\text{CM-Grad}}$,	*PCA*	Φ_{PCA},
Dispersion	Φ_{Dis},	*y-Distribution*	$\Phi_{\text{y-D}}$.
Information Content	Φ_{Inf},		

Feature sets requiring additional evaluations of the objective function (*Convexity*, *Local Search*, and *Curvature*) and cell-mapping feature sets with high computational or memory requirements in higher dimension or with the condition on the number of points in each cell (*Generalized CM*, *Barrier Trees*, and *Linear Model*) were omitted. *SOO Tree* features were published by Derbel et al. (2019) a year after our research published in (Pitra et al., 2018b), thus, they could not be taken into considerations.

Analysis of Results

We analyze relationships between the MSE of 29 different setting of GP or RF as described in the previous subsection and 79 landscape features, which were those from the 91 features mentioned in the previous subsection that didn't yield constant over all tested combinations of COCO noiseless functions and their instances for any for the tested GP or RF settings.

CORRELATION ANALYSIS We begin the analysis of results by a simple correlation analysis using the Spearman correlation coefficient between the MSE of the considered models. The reason for using this coefficient rather than the more common Pearson correlation coefficient is that the Pearson correlation coefficient quantifies linear dependence between both variables, whereas the Spearman correlation coefficient quantifies general monotone dependence.

No single landscape feature was found to be discriminative for surrogate model performance, although certain features from Φ_{Basic}, Φ_{Inf}, Φ_{MM}, and Φ_{PCA} are positively (or negatively) correlated with all considered models, which indicates the landscape to be difficult (or easy) for fitting any of them.

INFLUENCE OF LANDSCAPE FEATURES ON MODEL PERFORMANCE We have performed a multivariate statistical analysis using a classification trees.

For each of the 1080 different combinations of 3 dimensions, 24 noiseless functions and 15 their instances, the values of the 79 above mentioned non-constant landscape features were recorded. They were then classified into 29 classes according to which of the 29 considered settings of GP and RF was evaluated for the respective combination of dimension and noiseless function and achieved the lowest MSE among all evaluated settings. The classification tree trained on those data is the MATLAB implementation of the CART (Breiman, 1984). The resulting classification tree is depicted in Figure 38. It reveals that the most accurate model was nearly always either a GP or a boosted RF. Most frequently, the highest accuracy was achieved by a GP with the squared exponential covariance function κ_{SE}.

Features describing the global structure of the objective function landscape were detected as most distinctive

As most distinctive features were detected those describing the global structure of the objective function landscape, two CM features ($\varphi^{\text{CM-Angl}}_{\texttt{y\_best2worst\_std}}$ and $\varphi^{\text{CM-Conv}}_{\texttt{convex\_soft}}$), three Φ_{MM} features ($\varphi^{\text{MM}}_{\texttt{lin\_w\_interact\_adj\_r2}}$, $\varphi^{\text{MM}}_{\texttt{quad\_simple\_adj\_r2}}$, and $\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$), and two Φ_{NBC} features ($\varphi^{\text{NBC}}_{\texttt{nb\_std\_ratio}}$, $\varphi^{\text{NBC}}_{\texttt{nb\_mean\_ratio}}$). Global structure of the landscape can possibly influence the performance of a particular model. Very interesting is the discovered importance of basic features such as dimension $\varphi_{\texttt{dim}}$ or extreme values of the objective function $\varphi_{\texttt{objective\_min}}$. In addition, skewness $\varphi^{\text{y-D}}_{\texttt{skewness}}$ and the kurtosis $\varphi^{\text{y-D}}_{\texttt{kurtosis}}$ also had influence on surrogate model performance. The last mentioned observations may suggest that only a set of simple features can provide valuable information about the model performance.

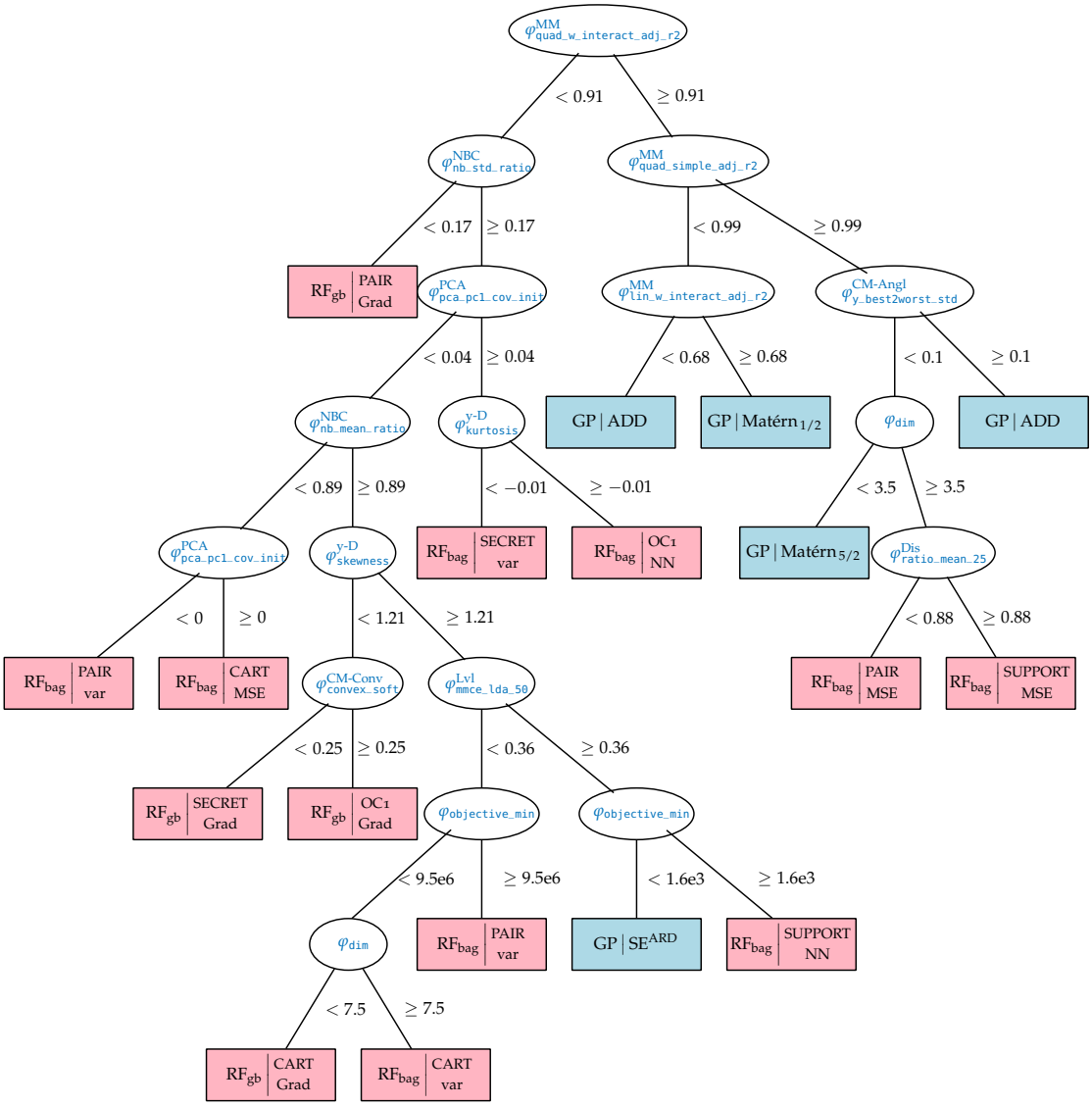


Figure 38: Classification tree demonstrating the influence of landscape features on model performance. Light blue: Gaussian processes. Light pink: Random forests. The RF models use the notation $\text{RF}_{\text{ensemble method}}\,\big|\,{\text{split method}\atop\text{error gain}}$, where bag = bagging, gb = gradient boosting, and Grad = gradient error gain. The GP models use the notation GP | covariance function.

5.1.2 *Conclusion*

This section investigated the relationships between two surrogate model types and features describing landscape of continuous fitness in static settings where the data are once sampled before the run of any single-objective black-box optimizer. The relationships between the performances of GP and RF surrogates with various settings and 11 landscape feature sets were analysed.

The results suggest that clear relationships between the performance of the 29 compared settings of GP and RF models and the considered features are not easy to derive. The features describing global properties of the landscape are highly influential on surrogate model settings performance. On the other hand, simple features can also provide important knowledge.

5.2 LANDSCAPE FEATURES INVESTIGATION

Our research in (Pitra et al., 2018b) and especially in (Pitra et al., 2019b), suggested that the investigation into properties of landscape features in the surrogate modeling context could bring more understanding into the entire problematics. Moreover, the analysis of landscape features also had less attention so far than it deserves only in the context of black-box optimization using static settings only, where the model is selected once at the beginning of the optimization process, and particularly not in the context of surrogate models (Renau et al., 2019, 2020).

In this section, we study properties of features representing the fitness landscape in the context of the data from actual runs of a surrogate-assisted version of the CMA-ES on the noiseless part of the COCO benchmark function testbed (Hansen et al., 2021), i. e., using data not presampled and respecting distribution and evaluation strategy of such CMA-ES version at the same time. From the large number of available features, we select a small number of representatives for subsequent research presented in Section 5.3, based on their robustness to sampling and similarity to other features.

First, we state the problem and research question connected with the relations between surrogate models and landscape features. Then, we present our set of new landscape features based on the CMA-ES state variables. Afterwards, we investigate the properties of landscape features in the context of the training set selection methods and select the most convenient features for the research in the next section.

5.2.1 *Problem Statement*

The problem can be formalized as follows: In a generation g of a surrogate-assisted version of the CMA-ES, a set of surrogate models $\mathcal{M}$ with settings ψ are trained utilizing particular choices of the training set $\mathcal{T}$. The training set $\mathcal{T}$ is selected out of an archive $\mathcal{A}$ ($\mathcal{T} \subset \mathcal{A}$) containing all points in which the fitness has been evaluated so far, using some TSS method (see Section 4.1.1). Afterwards, $\mathcal{M}$ is tested on the set of points sampled using the CMA-ES distribution $\mathbf{X}_{\text{te}} = \left\{\mathbf{x}_k \middle| \mathbf{x}_k \sim \mathcal{N}\left(\mathbf{m}^{(g)}, {\sigma^{(g)}}^2 \mathbf{C}^{(g)}\right), k = 1, \ldots, \alpha\right\}$, where $\alpha \in \mathbb{N}$ depends on the evolution control.

The research question connected to this problem is:

> What relationships between the suitability of different models for predicting the fitness and the considered landscape features do the testing results indicate?

5.2.2 *CMA-ES Landscape Features*

To utilize additional information comprised in CMA-ES state variables, we have proposed a set of features based on the CMA-ES in (Pitra et al., 2019b).

In each CMA-ES generation g during the fitness evaluation step (Step 4), the following additional features φ are obtained on the set of points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$:

- ◇ **Generation number** $\varphi^{\text{CMA}}_{\texttt{generation}} = g$ indicates the phase of the optimization process.
- ◇ **Step-size** $\varphi^{\text{CMA}}_{\texttt{step\_size}} = \sigma^{(g)}$ provides an information about the extent of the approximated region.
- ◇ **Number of restarts** $\varphi^{\text{CMA}}_{\texttt{restart}} = n_{\text{r}}^{(g)}$ performed till generation g may indicate landscape difficulty.
- ◇ **Mahalanobis mean distance** of the CMA-ES mean $\mathbf{m}^{(g)}$ to the sample mean $\mu_{\mathbf{X}}$ of $\mathbf{X}$

$$\varphi^{\text{CMA}}_{\texttt{mean\_dist}}(\mathbf{X}) = \sqrt{(\mathbf{m}^{(g)} - \mu_{\mathbf{X}})^{\top} \mathbf{C}_{\mathbf{X}}^{-1} (\mathbf{m}^{(g)} - \mu_{\mathbf{X}})}\,, \tag{103}$$

  where $\mathbf{C}_{\mathbf{X}}$ is the sample covariance of $\mathbf{X}$. This feature indicates suitability of $\mathbf{X}$ for model training from the point of view of the current state of the CMA-ES algorithm.

- ◇ Square of the $\mathbf{p}_c$ **evolution path** length $\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}} = \left\|\mathbf{p}_c^{(g)}\right\|^2$ is the only possible non-zero eigenvalue of *rank-one update* covariance matrix $\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^{\top}}$ (see Subsection 2.2.2). That feature providing information about the correlations between consecutive CMA-ES steps indicates a similarity of function landscapes among subsequent generations.

- ◇ $\mathbf{p}_\sigma$ **evolution path** ratio, i. e., the ratio between the evolution path length $\left\|\mathbf{p}_\sigma^{(g)}\right\|$ and the expected length of a random evolution path used to update step-size. It provides a useful information about distribution changes:

$$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}} = \frac{\left\|\mathbf{p}_\sigma^{(g)}\right\|}{E\left\|\mathcal{N}(\mathbf{0}, \mathbf{I})\right\|} = \frac{\left\|\mathbf{p}_\sigma^{(g)}\right\| \Gamma\left(\frac{D}{2}\right)}{\sqrt{2}\,\Gamma\left(\frac{D+1}{2}\right)}\,. \tag{104}$$

- ◇ **CMA similarity likelihood**. The log-likelihood of the set of points $\mathbf{X}$ with respect to the CMA-ES distribution may also serve as a measure of its suitability for training

$$\begin{aligned}\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathbf{X}) = &-\frac{N}{2}\left(D \ln 2\pi {\sigma^{(g)}}^2 + \ln\det \mathbf{C}^{(g)}\right) \\ &-\frac{1}{2}\sum_{\mathbf{x}\in\mathbf{X}} \left(\frac{\mathbf{x} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)^{\top} {\mathbf{C}^{(g)}}^{-1} \left(\frac{\mathbf{x} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)\,.\end{aligned} \tag{105}$$

For the completeness of feature definitions in Appendix A, we list CMA features also in Section A.5.

5.2.3 *Experimental Investigation of Landscape Features*

Based on the studies in (Bajer et al., 2019) and (Pitra et al., 2021), we have selected the DTS-CMA-ES as a successful representative of surrogate-assisted CMA-ES algorithms.

Investigation Settings

To investigate landscape features in the context of surrogate-assisted optimization, from the model point of view, we have decided to generate a large set of data using actual runs of an optimization algorithm on well-known benchmarks (similarly to (Renau et al., 2020; Saini et al., 2019)), more precisely, we have created 100 new runs from each performed run through CMA-ES distribution smoothing. In detail, we have generated[2] a dataset $\mathcal{D}$ of sample sets using independent runs of the 8 model settings from (Pitra et al., 2019b) for the DTS-CMA-ES algorithm (Bajer et al., 2019; Pitra et al., 2016) on the 24 noiseless single-objective benchmark functions from the COCO framework (Hansen et al., 2009b, 2021). All runs were performed in dimensions 2, 3, 5, 10, and 20 on instances $11-15$. The algorithm runs were terminated if the target fitness value 10^{-8} was reached or the budget of 250 function evaluations per dimension was depleted. Taking into account the double model training in DTS-CMA-ES (see Algorithm 9) and a large number of resulting datasets (120000), we have extracted archives $\mathcal{A}$ and testing sets $\mathbf{X}_{\text{te}}$ only from the first model training. The DTS-CMA-ES was employed in the overall best non-adaptive settings from (Bajer et al., 2019). To obtain 100 comparable archives and testing sets for landscape features investigation, we have generated a new collection of sample sets $\mathcal{D}_{100}$, where points for new archives and new populations are created using the weighted sum of original archive distributions from $\mathcal{D}$. The g-th generated dataset uses the weight vector $\mathbf{w}^{(g)} = \frac{1}{9}(0,\ldots,\underset{g-3}{0},\underset{g-2}{1},\underset{g-1}{2},\underset{g}{3},\underset{g+1}{2},\underset{g+2}{1},\underset{g+3}{0},\ldots,0)^\top$, which provides distribution smoothing across the available generations[3]. The data from well-known benchmarks were also used by Saini et al. (2019) and Renau et al. (2020).

From TSS methods in Subsection 4.1.1, we have selected the following three to be investigated: TSS full, TSS nearest, and TSS knn. Because each of these methods results in a different training set $\mathcal{T}$ using identical $\mathcal{A}$, we have performed all the feature investigations for each TSS method separately. By combining the two *basic sample sets* for feature calculation $\mathcal{A}$ and $\mathcal{T}$ with a population $\mathcal{P}$ consisting of the points without a known value of the original fitness to be evaluated by the surrogate model, we have obtained two new sets $\mathcal{A}_\mathcal{P} = \mathcal{A} \cup \mathcal{P}$ and $\mathcal{T}_\mathcal{P} = \mathcal{T} \cup \mathcal{P}$. Step 4 of Algorithm 5 performing the transformation into the $\sigma^2\mathbf{C}$ basis could also influence the landscape features. Thus, we have utilized either transformed and non-transformed sets for feature calculations, resulting in 8 different sample sets (4 in case of TSS full due to $\mathcal{T} = \mathcal{A}$): $\mathcal{A}$, $\mathcal{A}^\top$, $\mathcal{A}_\mathcal{P}$, $\mathcal{A}_\mathcal{P}^\top$, $\mathcal{T}$, $\mathcal{T}^\top$, $\mathcal{T}_\mathcal{P}$, $\mathcal{T}_\mathcal{P}^\top$, where $^\top$ denotes transformation into the $\sigma^2\mathbf{C}$ basis.

2 Source codes covering all mentioned datasets generation and experiments are available on https://github.com/bajeluk/surrogate-cmaes/tree/meta.

3 The weighted sum of the original archive distributions satisfies $\sum_{n=0}^{g_{\max}} w_n^{(g)} \mathcal{N}(\mathbf{m}^{(n)}, \mathbf{C}^{(n)}) = \mathcal{N}\big(\sum_{n=0}^{g_{\max}} w_n^{(g)} \mathbf{m}^{(n)}, \sum_{n=0}^{g_{\max}} (w_n^{(g)})^2 \mathbf{C}^{(n)}\big)$, where $g_{\max}$ is the maximal generation reached by the considered original archive and $\mathbf{m}^{(n)}$ and $\mathbf{C}^{(n)}$ are the mean and covariance matrix in CMA-ES generation n.

From each generated run in $\mathcal{D}_{100}$, we have uniformly selected 100 generations. In each such generations, we have computed all features from the following feature sets for all 3 considered TSS methods on all sample sets:

Basic	Φ_{Basic},	*Levelset*	Φ_{Lvl},
CMA features	Φ_{CMA},	*Meta-Model*	Φ_{MM},
Dispersion	Φ_{Dis},	*Nearest Better Clustering*	Φ_{NBC},
Information Content	Φ_{Inf},	*y-Distribution*	$\Phi_{\text{y-D}}$.

These sets do not require additional evaluations of the objective function and can be computed even in higher dimensions. (unlike e. g., *Convexity* or CM features). Some features were not computed due to the following reasons (see Appendix A for feature definitions): From Φ_{Basic}, we have used only $\varphi_{\texttt{dim}}$ and $\varphi_{\texttt{obs}}$ because the remaining ones were constant on the whole $\mathcal{D}_{100}$ dataset. $\varphi_{\texttt{dim}}$ is identical regardless the sample set. $\varphi_{\texttt{obs}}$ and all features from $\Phi_{\text{y-D}}$ were not computed in the $\sigma^2\mathbf{C}$ basis because it does not influence their resulting values. The feature $\varphi^{\text{MM}}_{\texttt{lin\_simple\_intercept}} \in \Phi_{\text{MM}}$ was excluded because it is useless if fitness normalization is performed (see Step 5 in Algorithm 5). The remaining features from Φ_{MM}, all the features from Φ_{NBC}, and all three features from $\Phi_{\text{y-D}}$ were not computed on sample sets with $\mathcal{P}$ because it also does not influence their resulting values.

For the rest of the paper, we will consider features which are independent of the sample set (i. e., $\varphi_{\texttt{dim}}$ and 5 Φ_{CMA} features) as a part of $\mathcal{A}$-based features only. This results in the total numbers of landscape features equal to 197 for TSS full (all were $\mathcal{A}$-based) and 388 for TSS nearest and TSS knn (from which 191 features were $\mathcal{T}$-based).

Feature Analysis Process and Its Results

The results of experiments concerning landscape features are presented in Tables 34–40.

First, we have investigated the impossibility of feature calculation (i. e., the feature value •) for each feature. Such information can be valuable and we will consider it as a valid output of any feature. On the other hand, large amount of • values on the tested dataset suggests low usability of the respective feature. Therefore, we have excluded features resulting in • in more than 25% of all measured values, which were for all TSS methods the feature $\varphi^{\text{Inf}}_{\texttt{eps\_s}}$ on $\mathcal{A}_{\mathcal{P}}$, $\mathcal{A}^{\top}_{\mathcal{P}}$, $\mathcal{T}_{\mathcal{P}}$, and $\mathcal{T}^{\top}_{\mathcal{P}}$ and for the TSS knn also the features $\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_02}}$, $\varphi^{\text{Dis}}_{\texttt{ratio\_median\_02}}$, $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}$, $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$ on $\mathcal{T}$, $\mathcal{T}^{\top}$, $\mathcal{T}_{\mathcal{P}}$, and $\mathcal{T}^{\top}_{\mathcal{P}}$, as well as $\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$ on $\mathcal{T}$ and $\mathcal{T}^{\top}$. This decreased the numbers of features to 195 for TSS full, 384 for TSS nearest, and 366 for TSS knn. Many features are difficult to calculate using low numbers of points. Therefore, for each feature we have measured the minimal number of points $N_{\bullet}$ in a particular combination of feature and sample set, for which the calculation resulted in • in at most 1% of cases. All measured values can be found in Tables 34–39. For the calculation of most of the features, the CMA-ES default population size value in $2D$: $N_{\bullet} = 4 + \lfloor 3 \ln 2 \rfloor = 6$, or initial point plus doubled default population size in $2D$: $N_{\bullet} = 1 + 2(4 + \lfloor 3 \ln 2 \rfloor) = 13$ was sufficient in all sample sets indexed with $\mathcal{P}$. Considering sample-set-independent features, no points are needed, because the values concern the CMA-ES iteration or CMA-ES run as a whole, not particular points. As can be seen from Tables 34–39, the dispersion features Φ_{Dis} require more points to be computable ($\varphi \neq \bullet$) as the quantile of function values used for splitting the sample set decreases (see Appendix A.6 for quantile values). $\varphi^{\text{MM}}_{\texttt{lin\_w\_interact\_adj\_r2}}$ and $\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$ have also high values of $N_{\bullet}$, which is plausible taking into account their descriptions (see Appendix A.9).

Table 34: TSS full features ($\mathcal{A} = \mathcal{T}$ for TSS full) from feature sets Φ_{Basic}, Φ_{CMA}, Φ_{Dis}, and $\Phi_{\text{y-D}}$. Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{A}$			$\mathcal{A}^\top$			$\mathcal{A}_\mathcal{P}$			$\mathcal{A}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi_{\texttt{dim}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi_{\texttt{obs}}$	1	100.00		1	100.00		13	100.00		13	100.00	
$\varphi^{\text{CMA}}_{\texttt{generation}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{restart}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}$	6	70.11		6	56.83		13	63.61		13	54.12	
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}$	1	99.75		1	99.96		13	99.75		13	99.96	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_02}}$	71	57.69	(2, 5), (3, 10), (3, 20)	71	62.33	(2, 5), (5, 20)	86	57.36	(2, 5), (3, 20)	86	62.16	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_02}}$	71	65.95	(3, 5)	71	69.28		86	65.41		86	69.60	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}$	71	47.75		71	99.52		86	48.77		86	99.52	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$	71	50.21		71	99.44		86	51.44		86	99.45	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_05}}$	27	55.22	(5, 20)	27	60.59		42	54.76	(5, 20)	42	60.32	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_05}}$	27	59.80		27	64.43		42	59.66	(5, 20)	42	64.57	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}$	27	47.99		27	99.49		42	49.16		42	99.49	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}$	27	51.16		27	99.46		42	52.88		42	99.48	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_10}}$	12	54.63	(5, 20)	12	59.33		25	54.06	(5, 20)	25	58.71	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_10}}$	12	58.29		12	62.38	(2, 3)	25	57.80		25	61.82	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}$	12	49.70		12	99.52		25	50.88		25	99.52	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}$	12	53.51		12	99.48		25	55.53		25	99.49	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_25}}$	6	54.80		6	58.86		15	53.87		15	58.06	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_25}}$	6	56.21		6	59.74		15	54.91		15	58.26	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}$	6	53.78		6	99.55		15	55.29		15	99.55	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}$	6	57.40		6	99.49		15	60.24		15	99.50	
$\varphi^{\text{y-D}}_{\texttt{skewness}}$	6	36.71		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{kurtosis}}$	6	63.06		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{number\_of\_peaks}}$	6	32.34		–	–	–	–	–	–	–	–	–

To increase comparability of investigated features, we normalize all the features to interval $[0, 1]$ using sigmoid function

$$\varphi_{\text{norm}}(x) = \frac{1}{1 + e^{-k(\varphi - \varphi_0)}}, \quad k = 2 \ln \frac{99}{Q_{0.99} - Q_{0.01}}, \quad \varphi_0 = \frac{Q_{0.01} + Q_{0.99}}{2}, \tag{106}$$

where $Q_{0.01}$ and $Q_{0.99}$ are 0.01 and 0.99 quantiles of feature φ on the whole $\mathcal{D}_{100}$ dataset considering values $\varphi \in \mathbb{R} \cup \{\pm\infty\}$. The normalization is derived to map feature quantiles to 0.01 and 0.99, i. e., $\varphi_{\text{norm}}(Q_{0.01}) = 0.01$ and $\varphi_{\text{norm}}(Q_{0.99}) = 0.99$. Such normalization maps infinity values to 0 and 1 and increases comparability of features with large differences in possible values (e. g., $\varphi^{\text{CMA}}_{\texttt{step\_size}} \in [1.5 \cdot 10^{-10}, 1.4 \cdot 10^{15}]$, whereas $\varphi^{\text{NBC}}_{\texttt{nb\_cor}} \in [-1, 1]$).

Table 35: TSS full features ($\mathcal{A} = \mathcal{T}$ for TSS full) from feature sets Φ_{Lvl}, Φ_{MM}, Φ_{Inf}, and Φ_{NBC}. Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{A}$			$\mathcal{A}^\top$			$\mathcal{A}_\mathcal{P}$			$\mathcal{A}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_10}}$	6	39.47		6	39.47		13	53.01		13	53.01	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_10}}$	6	69.30		6	69.88		13	65.30		13	65.91	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_10}}$	6	28.07		6	28.14		13	27.38		13	27.64	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}$	6	80.09		6	80.12		18	92.37		18	92.38	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_10}}$	6	44.32		6	42.71		13	46.42		13	48.00	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_10}}$	6	81.14		6	80.51		13	78.27		13	77.33	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_25}}$	6	32.64		6	32.64		13	54.19		13	54.19	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_25}}$	6	62.01		6	62.41		13	56.49		13	56.69	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_25}}$	6	22.86		6	22.63		13	27.03		13	26.99	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$	6	76.25		6	76.29	(3, 5)	15	94.13		15	94.12	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_25}}$	6	32.07	(3, 20)	6	27.68	(3, 10), (10, 20)	13	22.86		13	20.39	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_25}}$	6	77.24		6	77.68		13	63.08		13	63.11	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_50}}$	6	22.40		6	22.39		13	18.46		13	18.46	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_50}}$	6	48.24		6	49.46		13	32.88		13	33.15	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_50}}$	6	20.13		6	19.60		13	38.52		13	37.58	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_50}}$	6	69.22	(10, 20)	6	69.24	(10, 20)	15	84.45		15	84.41	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_50}}$	6	36.05		6	30.90		13	6.80	(5, 20)	13	4.51	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_50}}$	6	42.65	(3, 20)	6	40.68		13	21.96	(5, 20)	13	22.47	(3, 5), (5, 20)
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_adj\_r2}}$	6	19.64		6	19.62		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_min}}$	6	51.95		6	85.41		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max}}$	6	29.73		6	77.68		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max\_by\_min}}$	6	27.78		6	69.78		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_w\_interact\_adj\_r2}}$	100	44.77		100	43.89		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_adj\_r2}}$	6	50.23		6	52.16		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}$	6	46.45		6	95.21		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$	629	82.50		531	77.81		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{h\_max}}$	6	14.46	(3, 5)	6	33.83		13	35.74		13	35.42	
$\varphi^{\text{Inf}}_{\texttt{eps\_s}}$	6	6.59		6	34.65		3056	26.05		3196	54.56	
$\varphi^{\text{Inf}}_{\texttt{eps\_max}}$	6	64.93		6	84.99		13	65.17		13	84.18	
$\varphi^{\text{Inf}}_{\texttt{m0}}$	6	1.30		6	3.88		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{eps\_ratio}}$	6	27.67		6	44.91		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_std\_ratio}}$	6	31.11		6	18.27		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_mean\_ratio}}$	6	38.98		6	32.96		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_cor}}$	6	45.94		6	32.39		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{dist\_ratio}}$	6	55.23		6	50.78		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_fitness\_cor}}$	6	77.17		6	77.13		–	–	–	–	–	–

Table 36: TSS nearest features from feature sets Φ_{Basic}, Φ_{CMA}, Φ_{Dis}, and $\Phi_{\text{y-D}}$ (only $\mathcal{T}$-based and sample set indepent, $\mathcal{A}$-based are identical to TSS full in Table 34). Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{T}$			$\mathcal{T}^\top$			$\mathcal{T}_\mathcal{P}$			$\mathcal{T}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi_{\texttt{dim}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi_{\texttt{obs}}$	1	100.00		1	100.00		13	100.00		13	100.00	
$\varphi^{\text{CMA}}_{\texttt{generation}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{restart}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}$	6	70.11		6	56.83		13	63.61		13	54.12	
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}$	1	99.75		1	99.96		13	99.75		13	99.96	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_02}}$	71	57.69	(2, 5), (3, 10), (3, 20)	71	62.33	(2, 5), (5, 20)	86	57.36	(2, 5), (3, 20)	86	62.16	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_02}}$	71	65.95	(3, 5)	71	69.28		86	65.41		86	69.60	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}$	71	47.75		71	99.52		86	48.77		86	99.52	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$	71	50.21		71	99.44		86	51.44		86	99.45	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_05}}$	27	55.22	(5, 20)	27	60.59		42	54.76	(5, 20)	42	60.32	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_05}}$	27	59.80		27	64.43		42	59.66	(5, 20)	42	64.57	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}$	27	47.99		27	99.49		42	49.16		42	99.49	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}$	27	51.16		27	99.46		42	52.88		42	99.48	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_10}}$	12	54.63	(5, 20)	12	59.33		25	54.06	(5, 20)	25	58.71	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_10}}$	12	58.29		12	62.38	(2, 3)	25	57.80		25	61.82	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}$	12	49.70		12	99.52		25	50.88		25	99.52	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}$	12	53.51		12	99.48		25	55.53		25	99.49	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_25}}$	6	54.80		6	58.86		15	53.87		15	58.06	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_25}}$	6	56.21		6	59.74		15	54.91		15	58.26	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}$	6	53.78		6	99.55		15	55.29		15	99.55	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}$	6	57.40		6	99.49		15	60.24		15	99.50	
$\varphi^{\text{y-D}}_{\texttt{skewness}}$	6	36.71		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{kurtosis}}$	6	63.06		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{number\_of\_peaks}}$	6	32.34		–	–	–	–	–	–	–	–	–

We have tested the dependency of individual features on the dimension using feature medians from 100 samples for each distribution from $\mathcal{D}_{100}$. The Friedman's test rejected the hypothesis that the feature medians are independent of the dimension for all features at the family-wise significance level 0.05 using the Bonferroni-Holm correction. Moreover, for most of the features, the subsequently performed pairwise tests rejected the hypothesis of equality of feature medians for all pairs od dimensions. There were only several features from Φ_{Dis} and Φ_{Lvl} for which the hypothesis was not rejected for some pairs of dimensions (see Tables 34–39). Therefore, the influence of the dimension on the vast majority of features is essential.

Any analysis of the influence of multiple landscape features on the predictive error of surrogate models requires high robustness of features against random sampling of points. To have

Table 37: TSS nearest features from feature sets Φ_{Lvl}, Φ_{MM}, Φ_{Inf}, and Φ_{NBC} (only $\mathcal{T}$-based and sample set indepent, $\mathcal{A}$-based are identical to TSS full in Table 35). Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{T}$			$\mathcal{T}^\top$			$\mathcal{T}_\mathcal{P}$			$\mathcal{T}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_10}}$	6	39.47		6	39.47		13	53.01		13	53.01	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_10}}$	6	69.30		6	69.88		13	65.30		13	65.91	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_10}}$	6	28.07		6	28.14		13	27.38		13	27.64	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}$	6	80.09		6	80.12		18	92.37		18	92.38	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_10}}$	6	44.32		6	42.71		13	46.42		13	48.00	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_10}}$	6	81.14		6	80.51		13	78.27		13	77.33	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_25}}$	6	32.64		6	32.64		13	54.19		13	54.19	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_25}}$	6	62.01		6	62.41		13	56.49		13	56.69	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_25}}$	6	22.86		6	22.63		13	27.03		13	26.99	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$	6	76.25		6	76.29	(3, 5)	15	94.13		15	94.12	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_25}}$	6	32.07	(3, 20)	6	27.68	(3, 10), (10, 20)	13	22.86		13	20.39	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_25}}$	6	77.24		6	77.68		13	63.08		13	63.11	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_50}}$	6	22.40		6	22.39		13	18.46		13	18.46	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_50}}$	6	48.24		6	49.46		13	32.88		13	33.15	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_50}}$	6	20.13		6	19.60		13	38.52		13	37.58	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_50}}$	6	69.22	(10, 20)	6	69.24	(10, 20)	15	84.45		15	84.41	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_50}}$	6	36.05		6	30.90		13	6.80	(5, 20)	13	4.51	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_50}}$	6	42.65	(3, 20)	6	40.68		13	21.96	(5, 20)	13	22.47	(3, 5), (5, 20)
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_adj\_r2}}$	6	19.64		6	19.62		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_min}}$	6	51.95		6	85.41		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max}}$	6	29.73		6	77.68		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max\_by\_min}}$	6	27.78		6	69.78		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_w\_interact\_adj\_r2}}$	100	44.77		100	43.89		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_adj\_r2}}$	6	50.23		6	52.16		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}$	6	46.45		6	95.21		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$	629	82.50		531	77.81		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{h\_max}}$	6	14.46	(3, 5)	6	33.83		13	35.74		13	35.42	
$\varphi^{\text{Inf}}_{\texttt{eps\_s}}$	6	6.59		6	34.65		3056	26.05		3196	54.56	
$\varphi^{\text{Inf}}_{\texttt{eps\_max}}$	6	64.93		6	84.99		13	65.17		13	84.18	
$\varphi^{\text{Inf}}_{\texttt{m0}}$	6	1.30		6	3.88		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{eps\_ratio}}$	6	27.67		6	44.91		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_std\_ratio}}$	6	31.11		6	18.27		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_mean\_ratio}}$	6	38.98		6	32.96		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_cor}}$	6	45.94		6	32.39		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{dist\_ratio}}$	6	55.23		6	50.78		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_fitness\_cor}}$	6	77.17		6	77.13		–	–	–	–	–	–

Table 38: TSS knn features from feature sets Φ_{Basic}, Φ_{CMA}, Φ_{Dis}, and $\Phi_{\text{y-D}}$ (only $\mathcal{T}$-based and sample set indepent, $\mathcal{A}$-based are identical to TSS full in Table 34). Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{T}$			$\mathcal{T}^\top$			$\mathcal{T}_\mathcal{P}$			$\mathcal{T}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi_{\texttt{dim}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi_{\texttt{obs}}$	1	92.16		1	92.16		13	92.26		13	92.26	
$\varphi^{\text{CMA}}_{\texttt{generation}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{restart}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}$	6	78.64		6	98.40		13	93.96		13	98.71	
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	0	100.00		0	100.00		0	100.00		0	100.00	
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}$	1	98.81		1	99.95		13	98.80		13	99.96	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_02}}$	74	30.53		74	23.95		109	29.18		109	23.33	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_02}}$	74	34.49	(5, 10)	74	25.97		109	32.08		109	24.94	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}$	74	50.17		74	96.75		109	50.99		109	96.76	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$	74	49.55		74	95.54		109	50.37		109	95.47	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_05}}$	30	28.43		30	21.63		53	26.56		53	21.19	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_05}}$	30	35.85		30	24.73	(3, 10)	53	31.45	(2, 20)	53	23.58	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}$	30	55.94		30	96.80		53	56.94		53	96.85	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}$	30	55.11		30	95.80		53	56.23		53	95.82	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_10}}$	15	25.57	(2, 3), (2, 10), (3, 10), (5, 10)	15	19.13		28	24.33	(5, 10)	28	18.97	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_10}}$	15	32.86		15	21.96		28	29.68		28	21.49	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}$	15	58.33		15	96.96		28	59.42		28	97.02	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}$	15	57.49		15	96.09		28	58.85		28	96.16	
$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_25}}$	6	19.39		6	14.96		15	19.01		15	14.16	
$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_25}}$	6	23.38		6	15.76	(3, 5)	15	24.08		15	16.25	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}$	6	62.31		6	97.06		15	62.72		15	97.12	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}$	6	61.67		6	96.27		15	62.60		15	96.36	
$\varphi^{\text{y-D}}_{\texttt{skewness}}$	6	30.49		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{kurtosis}}$	6	72.61		–	–	–	–	–	–	–	–	–
$\varphi^{\text{y-D}}_{\texttt{number\_of\_peaks}}$	6	26.63		–	–	–	–	–	–	–	–	–

a robust set of k independent features, which return identical values for input in 95% cases, we would like all features to be identical in $100\sqrt[k]{0.95}$% cases. Thus, for our dataset $\mathcal{D}_{100}$ with 100 samples for each distribution even a small k requires all values to be identical, which is almost impossible to achieve for most of the investigated features. Therefore, we define feature *robustness* as a proportion of cases for which the difference between the 1st and 100th percentile calculated after standardization on samples from the same CMA-ES distribution is ≤ 0.05. Table 40 lists numbers of features achieving different levels of robustness. We have selected the robustness ≥ 0.9 to be used for subsequent analyses. The robustness calculated for individual features is listed in Tables 34–39. The chosen level of robustness excluded from further com-

Table 39: TSS knn features from feature sets Φ_{Lvl}, Φ_{MM}, Φ_{Inf}, and Φ_{NBC} (only $\mathcal{T}$-based, $\mathcal{A}$-based are identical to TSS full in Table 35). Features are grouped according to their feature sets (separated by horizontal lines). Features with less than 25% of values equal to • and robustness greater than 0.9, are in gray. $N_\bullet$ denotes the lowest measured number of points from which at most 1% of feature calculations resulted in • ($N_\bullet = 0$ for sample set independent φ). The (D_i, D_j) column shows the pairs of feature dimensions for which the two-sided Wilcoxon signed rank test with the Bonferroni-Holm correction does not reject the hypothesis of equality of median feature values, at the family-wise level 0.05 for each individual feature.

	$\mathcal{T}$			$\mathcal{T}^\top$			$\mathcal{T}_\mathcal{P}$			$\mathcal{T}_\mathcal{P}^\top$		
	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)	$N_\bullet$	rob.(%)	(D_i, D_j)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_10}}$	6	11.39		6	11.39		13	9.63		13	9.93	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_10}}$	6	67.83		6	69.40		13	69.58		13	70.79	(3, 5)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_10}}$	6	21.15		6	20.32		13	19.93		13	19.25	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}$	23	67.38		23	67.38		37	70.31		37	70.32	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_10}}$	6	15.74		6	13.69		13	22.17		13	20.80	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_10}}$	6	57.54		6	56.51		13	55.10		13	55.06	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_25}}$	6	9.73		6	9.63		13	15.42	(3, 20)	13	15.65	(3, 20)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_25}}$	6	37.27		6	38.00		13	36.70		13	37.25	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_25}}$	6	17.79		6	16.60		13	15.60		13	14.90	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$	6	73.93	(5, 10)	6	73.87	(5, 10)	18	89.97		18	89.95	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_25}}$	6	8.97		6	6.33		13	4.46		13	2.95	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_25}}$	6	47.47		6	49.72		13	35.90		13	39.50	(2, 10), (3, 5)
$\varphi^{\text{Lvl}}_{\texttt{mmce\_lda\_50}}$	6	17.46		6	16.62		13	7.86		13	7.88	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_qda\_50}}$	6	34.21		6	34.32		13	23.54		13	27.64	
$\varphi^{\text{Lvl}}_{\texttt{mmce\_mda\_50}}$	6	24.08		6	20.56		13	20.74		13	19.07	(2, 3)
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_50}}$	6	41.38		6	41.37		18	73.19		18	73.01	
$\varphi^{\text{Lvl}}_{\texttt{lda\_mda\_50}}$	6	26.79		6	32.00		13	2.70	(5, 10)	13	2.30	
$\varphi^{\text{Lvl}}_{\texttt{qda\_mda\_50}}$	6	17.30		6	24.22		13	5.95		13	7.69	(5, 20)
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_adj\_r2}}$	6	15.67		6	15.63		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_min}}$	6	97.40		6	63.93		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max}}$	6	98.12	(2, 3)	6	87.04		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max\_by\_min}}$	6	34.41		6	45.65		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{lin\_w\_interact\_adj\_r2}}$	100	37.50		100	30.25		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_adj\_r2}}$	6	41.50		6	37.91		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}$	6	92.55		6	87.21		–	–	–	–	–	–
$\varphi^{\text{MM}}_{\texttt{quad\_w\_interact\_adj\_r2}}$	688	69.48		632	60.02		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{n\_max}}$	6	1.94		6	1.57	(3, 5)	13	31.85		13	38.67	
$\varphi^{\text{Inf}}_{\texttt{eps\_s}}$	6	38.77		6	3.58		2424	66.99		2485	70.18	(5, 10)
$\varphi^{\text{Inf}}_{\texttt{eps\_max}}$	6	97.84		6	59.04		13	97.70		13	58.95	
$\varphi^{\text{Inf}}_{\texttt{m0}}$	6	0.84		6	0.44		–	–	–	–	–	–
$\varphi^{\text{Inf}}_{\texttt{eps\_ratio}}$	6	17.19		6	5.59		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_std\_ratio}}$	6	20.10		6	14.20		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_mean\_ratio}}$	6	36.38		6	30.42		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_cor}}$	6	28.49		6	26.50		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{dist\_ratio}}$	6	41.24		6	26.53		–	–	–	–	–	–
$\varphi^{\text{NBC}}_{\texttt{nb\_fitness\_cor}}$	6	45.12		6	45.14		–	–	–	–	–	–

Table 40: Proportion of features for individual TSS with robustness greater or equal to the threshold in the first column. The proportions in brackets represent $\mathcal{T}$-based features for given TSS (TSS full has only $\mathcal{A}$-based). The numbers in bold in the grey row are utilized for the following process.

threshold	TSS full	TSS nearest	TSS knn
0.5	**125**/195	**244**/384 (119/189)	**188**/366 (63/171)
0.6	**82**/195	**158**/384 (76/189)	**131**/366 (49/171)
0.7	**54**/195	**102**/384 (48/189)	**93**/366 (39/171)
0.8	**43**/195	**80**/384 (37/189)	**73**/366 (30/171)
0.9	**33**/195	**60**/384 (27/189)	**59**/366 (26/171)
0.99	**28**/195	**50**/384 (22/189)	**30**/366 (2/171)

putations all features from Φ_{NBC} and $\Phi_{\text{y}-\text{D}}$ for every TSS and all features from Φ_{Inf} for TSS full and TSS nearest. Both Φ_{NBC} and $\Phi_{\text{y}-\text{D}}$ are rather sensitive to the input data, where the influence of non-uniform sampling of data is probably not negligible. Features from Φ_{Inf} have very varied robustness. Whereas $\varphi^{\text{Inf}}_{\texttt{eps\_max}}$ provides high robustness around 0.85 on sample sets in the $\sigma^2\mathbf{C}$ basis (up to 0.97 for TSS knn), computations of $\varphi^{\text{Inf}}_{\texttt{m0}}$ and $\varphi^{\text{Inf}}_{\texttt{eps\_s}}$ resulted in the robustness 0.004 and 0.016, respectively, which were the two lowest among all features. The majority of Φ_{CMA} features provided high robustness caused mainly due to the independence of most of the features on the sample set. Features from Φ_{Lvl} based on quadratic discriminant analysis (`qda`) showed nearly double robustness compared to the rest of Φ_{Lvl} features. Transformation into the $\sigma^2\mathbf{C}$ basis increased robustness of specific types of features from Φ_{Dis} and Φ_{MM}. In particular, it increased the robustness of difference-based features from Φ_{Dis} to more than 0.99 and also of features based on model coefficients from Φ_{MM}. TSS knn is more sample dependent, therefore, the number of points in a sample set can vary. This also decreases the robustness of some ratio-based features from Φ_{Dis} or Φ_{Lvl}. On the other hand, coefficients of simple models from Φ_{MM} show robustness over 0.99 and $\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}$ mostly over 0.9. A noticeable dependence of robustness on which of the sets $\mathcal{A}$, $\mathcal{T}$, or $\mathcal{P}$ is used was not observed.

The large number of features suggests that for the purpose of investigation of their relationships to surrogate models, they should be clustered into a smaller number of groups of similar features. To this end, we have performed agglomerative hierarchical clustering according to the highest similarity $1 - \rho_{\text{SW}}(\varphi_i, \varphi_j)$, where ρ_{SW} is the correlation by Schweizer and Wolff (1981) and φ_i, φ_j are the vectors of all medians from $\mathcal{D}_{100}$ for the features i and j. To compensate for the ordering-dependency of agglomerative hierarchical clustering, we have performed 5 runs of clustering for each TSS method to find the optimal number of clusters. The number of clusters exceeding a threshold 0.9 for ρ_{SW}, averaged over all 15 runs, was 14. We have subsequently used that number as the value of k for subsequent k-medoids clustering using again Schweizer-Wolff correlation as a similarity. The features that are medoids of those 14 clusters are listed in Table 41. Even such a small number of feature representatives can be sufficient for achieving excellent performance in a subsequent investigation (Hoos et al., 2018; Renau et al., 2021). A majority of clusters contain features from the same feature set. Sometimes, the whole cluster is composed of the same features calculated only on different sample sets (e. g., $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$ on $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{A}^{\top}_{\mathcal{P}}$), which suggests that the influence of feature calculation on various sample sets might be negligible in those clusters. On the other hand, feature clusters for TSS knn often all share the base sample set ($\mathcal{A}$ or $\mathcal{T}$), or the same transformation even if the features

Table 41: Results of a medoid clustering into 14 clusters of the features for individual TSS methods, according to their Schweizer-Wolf correlation. The clusters are noted by numbers and separated by horizontal lines. Their medoid representatives are in gray.

	TSS full		TSS nearest		TSS knn
1	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_\mathcal{P})$	1	$\varphi^{\text{CMA}}_{\texttt{step\_size}}$		$\varphi^{\text{CMA}}_{\texttt{step\_size}}$
	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A})$
2	$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	2	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top)$
3	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A})$		$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{T}_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top)$		$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{T}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top)$	3	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top)$
	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top_\mathcal{P})$	1	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}_\mathcal{P})$
4	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{T}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{T}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{T}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{T}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top_\mathcal{P})$
5	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top)$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top)$		$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max}}(\mathcal{T})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}(\mathcal{T}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}(\mathcal{T}^\top_\mathcal{P})$
6	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top)$	4	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top)$
	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T})$	2	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top_\mathcal{P})$
	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{T}^\top)$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}^\top)$
7	$\varphi^{\text{CMA}}_{\texttt{restart}}$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{T}^\top)$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}^\top_\mathcal{P})$
8	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top)$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}_\mathcal{P})$	3	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T})$
	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{T}^\top_\mathcal{P})$		$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}_\mathcal{P})$
9	$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{T}^\top_\mathcal{P})$	4	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_\mathcal{P})$
10	$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top)$		$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}^\top_\mathcal{P})$
11	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{T}^\top)$
	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top_\mathcal{P})$	5	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{T}^\top)$
12	$\varphi_{\texttt{dim}}$	5	$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{A}^\top_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{T}^\top_\mathcal{P})$
13	$\varphi_{\texttt{obs}}(\mathcal{A})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{T}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_05}}(\mathcal{T}^\top_\mathcal{P})$
	$\varphi^{\text{CMA}}_{\texttt{generation}}$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{T}^\top)$	6	$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$
	$\varphi_{\texttt{obs}}(\mathcal{A}_\mathcal{P})$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{T}^\top_\mathcal{P})$	7	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_\mathcal{P})$
14	$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^\top)$		$\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{T}^\top_\mathcal{P})$		$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}^\top_\mathcal{P})$

Table 41: (continue)

	TSS nearest		TSS knn
6	$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	8	$\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_min}}(\mathcal{T})$
7	$\varphi_{\texttt{obs}}(\mathcal{A})$ $\varphi^{\text{CMA}}_{\texttt{generation}}$ $\varphi_{\texttt{obs}}(\mathcal{A}_\mathcal{P})$ $\varphi_{\texttt{obs}}(\mathcal{T})$ $\varphi_{\texttt{obs}}(\mathcal{T}_\mathcal{P})$		$\varphi^{\text{Inf}}_{\texttt{eps\_max}}(\mathcal{T})$ $\varphi^{\text{Inf}}_{\texttt{eps\_max}}(\mathcal{T}_\mathcal{P})$
8	$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top)$ $\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top_\mathcal{P})$ $\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}^\top)$ $\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}^\top_\mathcal{P})$	9	$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{T})$
9	$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	10	$\varphi^{\text{CMA}}_{\texttt{restart}}$
10	$\varphi_{\texttt{dim}}$	11	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{T}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{T}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{T}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_25}}(\mathcal{T}^\top_\mathcal{P})$
11	$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_\mathcal{P})$ $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}^\top_\mathcal{P})$ $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{T}_\mathcal{P})$ $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{T}^\top_\mathcal{P})$	12	$\varphi_{\texttt{obs}}(\mathcal{A})$ $\varphi^{\text{CMA}}_{\texttt{generation}}$ $\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^\top)$ $\varphi_{\texttt{obs}}(\mathcal{A}_\mathcal{P})$ $\varphi_{\texttt{obs}}(\mathcal{T})$ $\varphi_{\texttt{obs}}(\mathcal{T}_\mathcal{P})$
12	$\varphi^{\text{CMA}}_{\texttt{restart}}$		
13	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{T}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{T}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{T}^\top_\mathcal{P})$	13	$\varphi_{\texttt{dim}}$ $\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$ $\varphi^{\text{CMA}}_{\texttt{cma\_mean\_dist}}(\mathcal{T}_\mathcal{P})$
14	$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^\top)$ $\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{T}^\top)$	14	$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top)$ $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_02}}(\mathcal{A}^\top_\mathcal{P})$ $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top_\mathcal{P})$

are not from the same set of features. Considering the large numbers of available features, it is worth noticing that most of medoids are identical or at least very similar for all TSS methods. TSS full and TSS nearest medoids share identical features, where only 4 ($\varphi_{\texttt{obs}}$, $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$, $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$, $\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}$) differ in the sample set ($\mathcal{A}$ vs. $\mathcal{T}$, $\mathcal{A}^\top$ vs. $\mathcal{T}^\top$, $\mathcal{A}_\mathcal{P}$ vs. $\mathcal{T}_\mathcal{P}$, and $\mathcal{A}^\top$ vs. $\mathcal{T}^\top$ respectively). Notice that sample sets differ only in using $\mathcal{A}$ or $\mathcal{T}$. Moreover, 10 out of 14 representatives ($\varphi_{\texttt{dim}}$, $\varphi_{\texttt{obs}}$, $\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$, $\varphi^{\text{CMA}}_{\texttt{restart}}$, $\varphi^{\text{CMA}}_{\texttt{cma\_lik}}$, $\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}$, $\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}$, $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}$, $\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}$, $\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}$) are the same for all considered TSS methods, whereas sample sets utilized for feature calculation sometimes differ. Such similarity can indicate great importance of those features for characterizing the fitness landscape in the CMA-ES surrogate modeling context.

5.2.4 *Conclusion*

This section has presented an investigation of properties of features describing the fitness landscape in the context of surrogate-assisted evolutionary optimization by the CMA-ES for expensive continuous black-box tasks. Three sets of 14 landscape features were selected as valuable characteristics of the fitness landscape according to their properties, especially according to their robustness and similarity to other features. Up to our knowledge, this analysis of landscape feature properties in connection with surrogate models in evolutionary optimization, published in (Pitra et al., 2022), is first of its kind.

Most of the landscape features were designed to be calculated on some initial sample set from the more or less uniform distribution covering the whole search space to easily identify the optimized function and/or the algorithm with the highest performance on such function. In this section, we have operated on a more local level using only data from the actual runs of the optimization algorithm generated in each generation using Gaussian distribution, i. e., data completely different from the original design of those features. Therefore, the low number of robust and mostly similar features confirms the findings of Renau et al. (2020) that the distribution of the initial design has a notable impact on landscape analysis. The landscape features investigated in this section are in metalearning known as metafeatures.

The selected representatives of feature clusters are utilized in the following section to investigate the relationships between surrogate models and landscape features in the context of the identical problem defined in Section 5.2.1.

5.3 Landscape Analysis for Surrogate Models in the Evolutionary Black-Box Context

Following the research of landscape feature properties in the surrogate modeling context described in the previous section, we proceed by subsequential investigation of the relationships between the selected feature representatives and performances of surrogate models with various settings utilized during the run of the evolutionary optimizer on the black-box fitness function.

We have recently presented basic investigation in connection with landscape features of CMA-ES assisted by a surrogate model based on Gaussian processes in (Pitra et al., 2019b). In this section, we substantially more thoroughly investigate the relationships between selected representatives of landscape features and the error of several kinds of surrogate models using different settings of their parameters and the criteria for selecting points for their training. Such study, presented in (Pitra et al., 2022) is crucial due to completely different properties of data from runs of a surrogate-assisted algorithm compared with generally utilized sampling strategies, which imply different values of landscape features as emphasized in Renau et al. (2020).

5.3.1 *Relationships of Landscape Features and Surrogate Models*

To investigate the relationships between surrogate model errors and 14 landscape features selected for each of 3 TSS methods in the previous section, we have utilized 4 surrogate models: GPs, RFs, and two polynomial models from successful surrogate-assisted versions of the CMA-ES — lq model from lq-CMA-ES and lmm model from lmm-CMA (see Section 4.8.1 for

analysis of surrogate models and ECs utilized in these two algorithms). We have used the former two models in 8 and 9 different settings respectively. In (Saini et al., 2019), only Φ_{MM} features were utilized to investigate connection with several surrogate models in default settings in static scenario only, where the model is selected once for a specific problem at the beginning of the optimization process.

Experimental Settings

DATA We have split the dataset $\mathcal{D}$ constructed in Section 5.2.3 into validation and testing parts ($\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$) on the covariance function level uniformly at random (considering following levels of $\mathcal{D}$: dimension, function, instance, covariance function). More specifically, for each dimension, function, and instance in $\mathcal{D}$, we have uniformly selected runs using 7 covariance functions to $\mathcal{D}_{\text{test}}$ and 1 covariance to $\mathcal{D}_{\text{val}}$. In each of those runs, data from 25 uniformly selected generations were used.

ERROR MEASURES The two error measures utilized in our research each follow different aspects of model precision: The mean-squared error err_{MSE} (48) measures how much the model differs directly from the objective function landscape. On the other hand, the ranking difference error err_{RDE} (67) reflects that the CMA-ES state variables are adjusted according to the ordering of μ best points from the current population due to the invariance of the CMA-ES with respect to monotonous transformations.

TSS METHODS To select point for surrogate model training, two TSS methods were utilized: TSS full and TSS nearest. Considering the fact that TSS knn is the original TSS method of the lmm model from Kern et al. (2006), we also employ it but only in connection with this particular model.

LMM MODEL The regression model from lmm-CMA was used in its improved version published by Auger et al. (2013). The lmm model operates in the $\sigma^2\mathbf{C}$ basis in its own way, thus, the transformation step during training (step 4 in Algorithm 5) is not performed for this model.

LQ MODEL The linear-quadratic model was used in the version published by Hansen (2019). The original version utilizes all data without any transformation, therefore, the input data for lq model are not transformed for TSS full.

GP MODEL The GP regression model was employed in the version we have published in (Bajer et al., 2019) using 8 different covariance functions: κ_{LIN}, κ_{Q}, κ_{SE}, $\kappa_{\text{Mat}}^{5/2}$, κ_{RQ}, κ_{NN}, κ_{Gibbs}, and one example of composite function $\kappa_{\text{SE+Q}}$ (see covariance functions in Section 2.3.1).

RF MODEL The five splitting methods for decision tree from the following algoritms were employed: CART (Breiman, 1984), SECRET (Dobra and Gehrke, 2002), OC1 (Murthy et al., 1994), SUPPORT (Chaudhuri et al., 1994), and PAIR (Hinton and Revow, 1996). The settings of RF model for each combination of split method and error measure were found experimentally using latin-hypercube design on 100 out of 400 combinations of the number of trees in RF $n_{\text{tree}} \in \{2^6, 2^7, 2^8, 2^9, 2^{10}\}$, the number of points bootstrapped out of N training points $N_t \in \lceil \{1/4, 1/2, 3/4, 1\} \cdot N \rceil$, and the number of randomly subsampled dimensions used for training the individual tree $n_D \in \lceil \{1/4, 1/2, 3/4, 1\} \cdot D \rceil$. The maximum tree depth was set to 8, in

Table 42: Experimental settings of RF for 5 splitting methods found using latin-hypercube design on 100 out of 400 combinations of the number of trees in RF $n_{\text{tree}} \in \{2^6, 2^7, 2^8, 2^9, 2^{10}\}$, the number of points bootstrapped out of N training points $N_t \in \lceil \{1/4, 1/2, 3/4, 1\} \cdot N \rceil$, and the number of randomly subsampled dimensions used for training the individual tree $n_D \in \lceil \{1/4, 1/2, 3/4, 1\} \cdot D \rceil$. The winning settings on the validation dataset $\mathcal{D}_{\text{val}}$ are shown in the format $[n_{\text{tree}}, N_t, n_D]$ (in case of N_t and n_D are shown only the multipliers of N and D respectively).

TSS method	error	CART	SECRET	OC1	PAIR	SUPPORT
TSS full	MSE	$[2^{10}, 1, 3/4]$	$[2^8, 1/4, 1/2]$	$[2^7, 1, 1]$	$[2^7, 1, 3/4]$	$[2^7, 1, 1/4]$
	RDE	$[2^6, 3/4, 1]$	$[2^8, 3/4, 1]$	$[2^7, 1, 1]$	$[2^{10}, 3/4, 1]$	$[2^6, 3/4, 1]$
TSS nearest	MSE	$[2^8, 1, 3/4]$	$[2^9, 1/2, 1]$	$[2^7, 1, 1]$	$[2^7, 1, 3/4]$	$[2^{10}, 1/4, 1]$
	RDE	$[2^{10}, 3/4, 1]$	$[2^8, 3/4, 1]$	$[2^7, 1, 1]$	$[2^{10}, 3/4, 1]$	$[2^6, 3/4, 1]$

Table 43: Percentages of cases when the model did not provide usable prediction (model not trained, its prediction failed, or prediction is constant). SCRT = SECRET, SUPP = SUPPORT.

	GP								RF									lmm	lq
TSS	Gibbs	LIN	Mat	NN	Q	RQ	SE	SE+Q	CART MSE	CART RDE	SCRT MSE	SCRT RDE	OC1	PAIR MSE	PAIR RDE	SUPP MSE	SUPP RDE		
full	50.3	8.4	26.7	7.6	22.1	5.3	8.5	3.3	5.2	9.6	2.3	20.4	17.9	18.8	2.3	7.0	23.6	8.8	2.5
nearest	40.9	29.5	27.8	24.1	19.8	4.5	7.6	3.1	23.0	22.7	23.4	23.3	23.4	22.9	23.0	23.7	23.1	10.3	2.4
knn	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	5.5	—

accordance with (Chen and Guestrin, 2016). The remaining decision tree parameters have been taken identical to settings from (Pitra et al., 2018a) described in Section 4.7.1. The 9 winning settings on the validation dataset $\mathcal{D}_{\text{val}}$ are shown in Table 42. Preliminary testing showed that RFs performance in connection with the DTS-CMA-ES is higher when the input data are not transformed to the $\sigma^2\mathbf{C}$ basis. Thus, the transformation step is omitted during model training for both TSS.

Experimental Results

The results of analysing relationships between landscape features and two error measures for selected surrogate models with the appropriate settings are presented in Figures 39–41 and Tables 43–50.

We have summed up the cases when the model did not provide the prediction, i. e., the error value is not available, in Table 43. Such cases can occur when hyperparameters fitting fails, fitness prediction fails, or the model is considered constant (Steps 6, 8, and 10 in Algorithm 5 respectively). The numbers more or less confirm that more complex methods are more likely to fail. Whereas the lq model, the most simple model among all tested, provided predictions in almost 98% of all cases, where the missing results can be attributed to constant predictions, the GP model with Gibbs covariance function provided only 55% of the required predictions. Generally, the models were able to provide predictions more often when using TSS full opposite to TSS nearest, probably due to the locality of the training set, considering that it is easier to train a constant model on a smaller area, where the differences between objective values are more likely negligible. The TSS knn was the most successful selection method for lmm, probably because it was designed directly for this model. In the following investigation,

the cases where the error values were missing for a particular model, were excluded from all comparisons involving that model.

We have tested the statistical significance of the $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$ differences for 19 surrogate model settings using TSS full and TSS nearest methods and also the lmm surrogate model utilizing TSS knn, i. e., 39 different combinations of model settings ψ and TSS methods (ψ, TSS), on all available sample sets using the Friedman test (Demšar, 2006). The resulting p-values of the two Friedman tests, one for each error measure, are below the smallest double precision number. A pairwise comparison of the model settings with respect to $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$ revealed significant differences among the vast majority of pairs of model settings according to the non-parametric two-sided Wilcoxon signed rank test with the Holm correction for the family-wise error. To better illustrate the differences between individual settings, we also count the percentage of cases at which one setting had the error lower than the other. The pairwise score and the statistical significance of the pairwise differences are summarized in Tables 44 and 45. Results of statistical tests confirmed that the obtained $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$ values are sufficiently diverse for further investigation of model settings suitability. The best overall results were provided by GPs with all covariances except κ_{LIN}. Especially, GP using $\kappa_{\mathrm{SE+Q}}$ as a covariance function in TSS nearest significantly outperformed all other (ψ, TSS) combinations. The polynomial models using TSS nearest take the second place in such comparison, being outperformed only by the GP models mentioned above. Generally, models using TSS nearest provided better results than when using TSS full (in case of lmm also better than TSS knn). The percentages of $\mathrm{err}_{\mathrm{RDE}}$ show smaller differences in precision than $\mathrm{err}_{\mathrm{MSE}}$ due to the larger probability of error equality on the limited number of possible $\mathrm{err}_{\mathrm{RDE}}$ values compared to the infinite number of possible $\mathrm{err}_{\mathrm{MSE}}$ values.

To compare the convenience of individual features as descriptors of areas where the surrogate model $\mathcal{M}$ with a particular (ψ, TSS) combination has the best performance, we use the Kolmogorov-Smirnov test (KS test) testing the equality of the distribution of values of individual features calculated on the whole $\mathcal{D}_{\mathrm{test}}$ and on only those sample sets from $\mathcal{D}_{\mathrm{test}}$ for which the (ψ, TSS) combination leads to the lowest error ($\mathrm{err}_{\mathrm{MSE}}$ or $\mathrm{err}_{\mathrm{RDE}}$) among all tested combinations. The hypothesis of distribution equality is tested at the family-wise significance level $\alpha = 0.05$ after the Holm correction. The resulting p-values summarized in Tables 46–50 and visualised in color in Figure 39 show significant differences in distribution among combinations of model settings and TSS method (ψ, TSS) for the vast majority of considered features. Features from Φ_{Dis} and $\varphi^{\mathrm{CMA}}_{\texttt{step\_size}}$ provided the most significant differences for almost all models. For the GP models and the sample set leading to the lowest $\mathrm{err}_{\mathrm{MSE}}$, the p-values were often even below the smallest double precision number. These features also provided very low p-values for lmm and lq model. The only exception is $\varphi^{\mathrm{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{T}^{\top})$ providing notably higher p-values for all model settings. Moreover, features calculated on $\mathcal{T}$-based sample sets for TSS nearest provided in most cases higher p-values than when calculated on $\mathcal{A}$-based.

To perform a multivariate statistical analysis we have built two classification trees: one for TSS full and one for TSS nearest. The data for each TSS method described by 14 features selected in Section 5.2 were divided into 19 classes according to which of the 19 considered surrogate model settings achieved the lowest $\mathrm{err}_{\mathrm{RDE}}$. In case of a tie, the model with the lowest $\mathrm{err}_{\mathrm{MSE}}$ among models with the lowest $\mathrm{err}_{\mathrm{RDE}}$ was the chosen one. The classification tree trained on those data is the MATLAB implementation of the CART (Breiman, 1984), where the minimal number of cases in leaf was set to 5000, the twoing rule was used as a splitting criterion, $\varphi_{\texttt{dim}}$ was considered as a discrete variable and the remaining 13 features were considered as continuous. The resulting classification trees are depicted in Figures 40 and 41.

Table 44: A pairwise comparison of the models $\mathcal{M}$ with settings ψ for err_{MSE} in different TSS. The percentage of wins of i-th model setting against j-th model setting over all available data is given in the i-th row and j-th column. The numbers in bold mark the row model setting being significantly better than the column model setting according to the two-sided Wilcoxon signed rank test with the Holm correction at family-wise significance level $\alpha = 0.05$. SCRT = SECRET, SUPP = SUPPORT.

MSE	TSS	full																				nearest																knn
	$\mathcal{M}$	GP								RF									lmm	lq		GP								RF								lmm	lq	lmm
TSS	$\mathcal{M}$	ψ	Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			
full	GP	Gibbs		**85**	**68**	**53**	43	35	38	47	**73**	**72**	**77**	**78**	**77**	**78**	**58**	**71**	**62**	**70**	**70**	42	**69**	**52**	49	35	34	34	36	**62**	**61**	**54**	**63**	**61**	**63**	**57**	**59**	**60**	**51**	**51**	**59**
		LIN	15		13	6	10	10	5	10	11	11	14	12	13	15	8	10	9	9	11	12		11	8	7	6	5	5	5	11	10	9	10	10	11	10	10	11	10	8	16
		Mat	32	**87**		24	25	19	18	22	42	41	46	45	**52**	**54**	30	40	33	27	49	29	41		24	27	19	20	20	20	34	33	30	34	32	35	32	33	34	25	25	33
		NN	47	**94**	**76**		**50.2**	34	36	47	**77**	**80**	**78**	**75**	**87**	**88**	**52**	**75**	**59**	**80**	**81**	37	**58**	42	36		25	19	20	21	**53**	**52**	46	**55**	**52**	**56**	48	**54**	**52**	34	36	38
		Q	**57**	**90**	**75**	49.8		36	38	48	**66**	**67**	**69**	**69**	**73**	**75**	**54**	**65**	**59**	**63**	**68**	48	**61**	**52**	49	35		34	35	37	**57**	**57**	**52**	**58**	**57**	**59**	**54**	**57**	**57**	45	45	48
		RQ	**65**	**90**	**81**	**66**	**64**		**58**	**68**	**78**	**81**	**81**	**78**	**83**	**84**	**62**	**77**	**67**	**79**	**78**	46	**61**	**55**	49.6	37	29		31	**61**	**60**	**55**	**62**	**60**	**62**	**57**	**62**	**60**	42	42	44
		SE+Q	**62**	**95**	**82**	**64**	**62**	42		**69**	**83**	**85**	**83**	**80**	**90**	**91**	**62**	**81**	**67**	**84**	**84**	47	**61**	**53**	48	35	28	28		**61**	**60**	**55**	**62**	**60**	**63**	**57**	**62**	**60**	41	43	43
		SE	**53**	**90**	**78**	**53**	**52**	32	31		**74**	**77**	**77**	**73**	**81**	**81**	**55**	**73**	**61**	**77**	**77**	41	**56**	47	43	29	24	24	25		**54**	**54**	48	**55**	**53**	**56**	**51**	**55**	**53**	35	36	38
	RF	CART MSE	27	**89**	**58**	23	34	22	17	26		**50.02**	**57**	**55**	**72**	**74**	23	42	33	**59**	**63**	22	42	25	19	15	12	12	13	33	32	26	35	31	36	29	34	34	25	25	30
		CART RDE	28	**89**	**59**	20	33	19	15	23	49.98		**56**	**53**	**68**	**71**	22	42	31	**59**	**63**	24	42	25	19	15	12	12	14	33	32	25	34	30	35	29	32	34	25	25	31
		OC1	23	**86**	**54**	22	31	19	17	23	43	44		44	**61**	**64**	18	39	27	**56**	**59**	20	42	21	16	13	11	11	12	26	25	20	28	24	29	22	29	27	24	24	33
		PAIR MSE	22	**88**	**55**	25	31	22	20	27	45	47	**56**		**63**	**66**	19	41	26	**57**	**60**	17	47	22	15	12	9	9	11	26	26	19	28	24	29	22	29	27	24	23	33
		PAIR RDE	23	**87**	48	13	27	17	10	19	28	32	39	37		**57**	16	24	23	47	**52**	20	35	19	14	11	9	9	10	25	25	20	27	23	28	23	26	27	21	20	26
		SCRT MSE	22	**85**	46	12	25	16	9	19	26	29	36	34	43		15	22	22	44	49.7	19	34	18	13	11	8	9	9	24	23	19	26	22	26	22	24	26	20	19	26
		SCRT RDE	42	**92**	**70**	48	46	38	38	45	**77**	**78**	**82**	**81**	**84**	**85**		**75**	**58**	**73**	**75**	34	**67**	43	34	26	22	22	24	**54**	**53**	42	**58**	53	**59**	47	**55**	**54**	40	41	45
		SUPP MSE	29	**90**	**60**	25	35	23	19	27	**58**	**58**	**61**	**59**	**76**	**78**	25		37	**62**	**65**	24	45	27	20	16	13	13	14	37	36	28	39	35	39	32	38	37	26	27	32
		SUPP RDE	38	**91**	**67**	41	41	33	33	39	**67**	**69**	**73**	**74**	**77**	**78**	42	**63**		**69**	**71**	30	**64**	38	30	22	19	19	21	47	46	35	**50.4**	45	**51**	40	47	47	37	38	43
	lmm		30	**91**	**73**	20	37	21	16	23	41	41	44	43	**53**	**56**	27	38	31		**77**	27	37	27	24	19	18	18	19	30	29	27	30	29	31	28	30	30	24	27	32
	lq		30	**89**	**51**	19	32	22	16	23	37	37	41	40	48	**50.3**	25	35	29	23		27	35	24	23	19	17	17	17	29	28	26	29	28	30	27	28	29	20	20	26
nearest	GP	Gibbs	**58**	**88**	**71**	**63**	**52**	54	**53**	**59**	**78**	**76**	**80**	**83**	**80**	**81**	**66**	**76**	**70**	**73**	**73**		**79**	**63**	**58**	45	41	39	44	**72**	**71**	**64**	**73**	**71**	**74**	**67**	**69**	**70**	**62**	**63**	**70**
		LIN	31	**89**	**59**	42	39	39	39	44	**58**	**58**	**58**	**53**	**65**	**66**	33	**55**	36	**63**	**65**	21		20	15	13	9	9	10	25	24	17	25	22	25	19	27	23	21	25	29
		Mat	48	**92**	**76**	**58**	48	45	47	53	**75**	**75**	**79**	**78**	**81**	**82**	**57**	**73**	**62**	**73**	**76**	37	**80**		41	26	22	23	25	**58**	**57**	47	**61**	**56**	**62**	**51**	**57**	**56**	37	44	47
		NN	**51**	**93**	**73**	**64**	**51**	50.4	**52**	**57**	**81**	**81**	**84**	**85**	**86**	**87**	**66**	**80**	**70**	**76**	**77**	42	**85**	**59**		36	24	25	28	**71**	**70**	**60**	**75**	**70**	**76**	**63**	**68**	**69**	49	**54**	**54**
		Q	**65**	**94**	**81**	**75**	**65**	**63**	**65**	**71**	**85**	**85**	**87**	**88**	**89**	**89**	**74**	**84**	**78**	**81**	**81**	**55**	**87**	**74**	**64**		41	39	49	**77**	**76**	**70**	**80**	**76**	**80**	**71**	**76**	**75**	**60**	**62**	**64**
		RQ	**66**	**95**	**80**	**81**	**66**	**71**	**72**	**76**	**88**	**88**	**89**	**91**	**91**	**92**	**78**	**87**	**81**	**82**	**83**	**59**	**91**	**78**	**76**	**59**		48	53	**83**	**83**	**78**	**85**	**83**	**86**	**79**	**83**	**81**	**67**	**74**	**72**
		SE+Q	**66**	**95**	**80**	**80**	**65**	**70**	**72**	**76**	**88**	**88**	**89**	**91**	**91**	**91**	**78**	**87**	**81**	**82**	**83**	**61**	**91**	**77**	**75**	**61**	**52**		**61**	**83**	**83**	**78**	**85**	**82**	**86**	**79**	**82**	**81**	**68**	**74**	**72**
		SE	**64**	**95**	**80**	**79**	**63**	**69**	**69**	**75**	**87**	**86**	**88**	**89**	**90**	**91**	**76**	**86**	**79**	**81**	**83**	**56**	**90**	**75**	**72**	**51**	47	39		**81**	**80**	**74**	**83**	**80**	**83**	**76**	**80**	**79**	**65**	**72**	**69**
	RF	CART MSE	38	**89**	**66**	47	43	39	39	46	**67**	**67**	**74**	**74**	**75**	**76**	46	**63**	53	**70**	**71**	28	**75**	42	29	23	17	17	19		46	33	**56**	45	**58**	39	49	47	37	39	45
		CART RDE	39	**90**	**67**	48	43	40	40	46	**68**	**68**	**75**	**74**	**75**	**77**	47	**64**	**54**	**71**	**72**	29	**76**	43	30	24	17	17	20	54		34	**58**	48	**60**	40	49.8	49	38	40	46
		OC1	46	**91**	**70**	**54**	48	45	45	**52**	**74**	**75**	**80**	**81**	**80**	**81**	**58**	**72**	**65**	**73**	**74**	36	**83**	**53**	40	30	22	22	26	**67**	**66**		**72**	**65**	**73**	**57**	**66**	**65**	46	47	**51**
		PAIR MSE	37	**90**	**66**	45	42	38	38	45	**65**	**66**	**72**	**72**	**73**	**74**	42	**61**	49.6	**70**	**71**	27	**75**	39	25	20	15	15	17	44	42	28		40	**52**	33	43	42	36	37	43
		PAIR RDE	39	**90**	**68**	48	43	40	40	47	**69**	**70**	**76**	**76**	**77**	**78**	47	**65**	**55**	**71**	**72**	29	**78**	44	30	24	17	18	20	**55**	**52**	35	**60**		**62**	41	**51**	**50.2**	39	40	46
		SCRT MSE	37	**89**	**65**	44	41	38	37	44	**64**	**65**	**71**	**71**	**72**	**74**	41	**61**	49	**69**	**70**	26	**75**	38	24	20	14	14	17	42	40	27	48	38		32	43	41	35	37	43
		SCRT RDE	43	**90**	**68**	**52**	46	43	43	49	**71**	**71**	**78**	**78**	**77**	**78**	**53**	**68**	**60**	**72**	**73**	33	**81**	49	37	29	21	21	24	**61**	**60**	43	**67**	**59**	**68**		**58**	**60**	42	44	49
		SUPP MSE	41	**90**	**67**	46	43	38	38	45	**66**	**68**	**71**	**71**	**74**	**76**	45	**62**	**53**	**70**	**72**	31	**73**	43	32	24	17	18	20	**51**	**50.2**	34	**57**	49	**57**	42		49.9	39	40	45
		SUPP RDE	40	**89**	**66**	48	43	40	40	47	**66**	**66**	**73**	**73**	**73**	**74**	46	**63**	**53**	**70**	**71**	30	**77**	44	31	25	19	19	21	**53**	**51**	35	**58**	49.8	**59**	40	**50.1**		39	40	46
	lmm		49	**90**	**75**	**66**	**55**	58	59	**65**	**75**	**75**	**76**	**76**	**79**	**80**	**60**	**74**	**63**	**76**	**80**	38	**79**	**63**	**51**	40	33	32	35	**63**	**62**	**54**	**64**	**61**	**65**	**58**	**61**	**61**		**67**	**58**
	lq		49	**92**	**75**	**64**	**55**	**58**	**57**	**64**	**75**	**75**	**76**	**77**	**80**	**81**	**59**	**73**	**62**	**73**	**80**	37	**75**	**56**	46	38	26	26	28	**61**	**60**	**53**	**63**	**60**	**63**	**56**	**60**	**60**	33		47
knn	lmm		41	**84**	**67**	**62**	**52**	**56**	**57**	**62**	**70**	**69**	**67**	**67**	**74**	**74**	**55**	**68**	**57**	**68**	**74**	30	**71**	**53**	46	36	28	28	31	55	54	49	**57**	54	**57**	**51**	**55**	54	42	**53**	

Table 45: A pairwise comparison of the models $\mathcal{M}$ with settings ψ for err_{RDE} in different TSS. The percentage of wins of i-th model setting against j-th model setting over all available data is given in the i-th row and j-th column. The numbers in bold mark the row model setting being significantly better than the column model setting according to the two-sided Wilcoxon signed rank test with the Holm correction at family-wise significance level $\alpha = 0.05$. SCRT = SECRET, SUPP = SUPPORT.

RDE		TSS	full																			nearest																knn
		$\mathcal{M}$	GP								RF								lmm	lq		GP							RF								lmm	lq	lmm
TSS	$\mathcal{M}$	ψ	Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			
full	GP	Gibbs		**76**	**63**	**55**	47	42	44	48	**63**	**62**	**67**	**67**	**67**	**67**	58	**62**	58	62	62	42	**74**	**60**	**55**	42	39	39	41	**60**	**60**	**56**	**63**	**60**	**63**	59	57	**61**	50.03	51	50.1
		LIN	24		34	29	26	24	26	27	35	32	34	33	39	40	25	33	24	36	36	17	41	25	19	16	10	10	11	25	25	21	28	24	28	24	22	26	17	18	17
		Mat	37	**66**		40	33	31	32	34	48	47	**52**	**51**	**53**	**54**	42	47	41	43	47	32	**62**	40	37	27	24	24	25	43	43	40	47	43	47	42	41	45	31	33	33
		NN	45	**71**	**60**		43	37	39	42	**58**	**57**	**61**	**61**	**64**	**64**	51	**57**	**51**	**57**	**57**	35	**68**	51	42	33	23	23	25	**53**	**53**	48	**57**	**53**	**57**	**52**	**51**	**54**	34	37	34
		Q	**53**	**74**	**67**	**57**		44	46	**51**	**62**	**61**	**65**	**65**	**67**	**67**	**56**	**61**	**56**	**62**	**63**	44	**72**	**58**	**51**	39	35	34	36	**58**	**57**	**54**	**61**	**58**	**61**	**57**	**56**	**59**	45	46	44
		RQ	**58**	**76**	**69**	**63**	**56**		**54**	**57**	**65**	**64**	**68**	**68**	**70**	**70**	**60**	**64**	**59**	**65**	**65**	44	**74**	**62**	**53**	44	30	31	33	**61**	**61**	**57**	**64**	**61**	**64**	**60**	**59**	**62**	42	45	41
		SE+Q	**56**	**74**	**68**	**61**	**54**	46		**57**	**64**	**63**	**67**	**66**	**69**	**69**	**59**	**62**	**58**	**63**	**63**	43	**72**	**61**	**52**	43	30	29	31	**60**	**60**	**56**	**63**	**60**	**63**	**59**	**58**	**61**	40	43	40
		SE	**52**	**73**	**66**	**58**	49	43	43		**62**	**60**	**65**	**64**	**67**	**68**	**55**	**61**	**55**	**62**	**62**	40	**71**	**58**	49	38	28	26	27	**57**	**57**	**52**	**60**	**57**	**61**	**56**	**54**	**58**	38	41	37
	RF	CART MSE	37	**65**	**52**	42	38	35	36	38		48	**55**	**53**	**58**	**59**	40	48	41	**51**	**51**	30	**61**	42	34	29	20	21	21	44	44	38	48	43	49	42	41	46	30	33	30
		CART RDE	38	**68**	**53**	43	39	36	37	40	**52**		**57**	**55**	**60**	**60**	41	49.8	41	**53**	**53**	30	**63**	43	36	29	22	22	23	44	44	39	49	44	49.7	43	42	47	32	35	32
		OC1	33	**66**	48	39	35	32	33	35	45	43		48	**53**	**54**	35	43	34	48	49	27	**60**	39	31	26	20	20	21	37	37	33	43	37	43	36	36	39	30	32	30
		PAIR MSE	33	**67**	49	39	35	32	34	36	47	45	**52**		**55**	**55**	35	45	34	49.6	**50.2**	27	**63**	40	31	26	19	19	20	39	39	33	45	38	45	37	36	41	30	32	30
		PAIR RDE	33	**61**	47	36	33	30	31	33	42	40	47	45		**51**	33	40	33	45	46	26	**56**	37	28	24	16	16	17	36	36	31	42	35	42	35	34	39	26	28	26
		SCRT MSE	33	**60**	46	36	33	30	31	32	41	40	46	45	49		32	39	32	45	45	26	**56**	36	27	23	16	16	17	35	35	30	40	34	41	34	33	38	25	28	25
		SCRT RDE	42	**75**	**58**	49	44	40	41	45	**60**	**59**	**65**	**65**	**67**	**68**		58	49.8	**59**	**59**	35	**71**	50.02	42	34	28	27	29	**53**	**52**	47	**58**	**52**	**58**	51	**50.3**	**55**	39	41	39
		SUPP MSE	38	**67**	**53**	43	39	36	38	39	**52**	**50.2**	**57**	**55**	**60**	**61**	42		42	**53**	**53**	31	**63**	43	35	29	22	22	22	46	45	40	49.9	45	**50.3**	44	42	48	32	34	32
		SUPP RDE	42	**76**	**59**	49	44	41	42	45	**59**	**59**	**66**	**66**	**67**	**68**	**50.2**	**58**		**60**	**60**	35	**73**	**51**	43	34	28	28	29	**53**	**53**	47	**59**	**53**	**59**	**52**	51	**56**	40	42	40
	lmm		38	**64**	**57**	43	38	35	37	38	49	47	**52**	50.4	**55**	**55**	41	47	40		**55**	32	**60**	44	37	30	23	23	23	43	42	38	46	42	46	41	40	44	28	31	29
	lq		38	**64**	**53**	43	37	35	37	38	49	47	**51**	49.8	**54**	**55**	41	47	40	45		32	**60**	43	37	30	22	22	22	42	42	39	46	42	46	41	40	43	27	29	28
nearest	GP	Gibbs	**58**	**83**	**68**	**65**	**56**	**56**	**57**	**60**	**70**	**70**	**73**	**73**	**74**	**74**	**65**	**69**	**65**	**68**	**68**		**80**	**67**	**63**	**52**	47	46	49	**68**	**68**	**65**	**71**	**67**	**70**	**67**	**66**	**68**	**59**	**60**	**60**
		LIN	26	**59**	38	32	28	26	28	29	39	37	40	37	44	44	29	37	27	40	40	20		26	21	17	12	12	13	29	29	25	32	28	32	27	26	30	20	22	21
		Mat	40	**75**	**60**	49	42	38	39	42	**58**	**57**	**61**	**60**	**63**	**64**	49.98	**57**	49	**56**	**57**	33	**74**		42	28	24	24	26	**52**	**51**	47	**56**	**51**	**56**	50.2	49	**53**	34	37	37
		NN	45	**81**	**63**	**58**	49	47	48	**51**	**66**	**64**	**69**	**69**	**72**	**73**	**58**	**65**	**57**	**63**	**63**	37	**79**	**58**		38	26	26	29	**61**	**60**	**56**	**65**	**60**	**66**	**59**	**59**	**61**	43	45	44
		Q	**58**	**84**	**73**	**67**	**61**	**56**	**57**	**62**	**71**	**71**	**74**	**74**	**76**	**77**	**66**	**71**	**66**	**70**	**70**	48	**83**	**72**	**62**		40	36	40	**68**	**68**	**65**	**72**	**68**	**72**	**67**	**67**	**69**	**53**	**55**	53
		RQ	**61**	**90**	**76**	**77**	**65**	**70**	**70**	**72**	**80**	**78**	**80**	**81**	**84**	**84**	**72**	**78**	**72**	**77**	**78**	**53**	**88**	**76**	**74**	**60**		47	**52**	**75**	**75**	**72**	**78**	**75**	**78**	**74**	**74**	**75**	**63**	**67**	**63**
		SE+Q	**61**	**90**	**76**	**77**	**66**	**69**	**71**	**74**	**79**	**78**	**80**	**81**	**84**	**84**	**73**	**78**	**72**	**77**	**78**	**54**	**88**	**76**	**74**	**64**	**53**		**62**	**75**	**75**	**72**	**78**	**75**	**78**	**74**	**74**	**75**	**64**	**67**	**64**
		SE	**59**	**89**	**75**	**75**	**64**	**67**	**69**	**73**	**79**	**77**	**79**	**80**	**83**	**83**	**71**	**78**	**71**	**77**	**78**	**51**	**87**	**74**	**71**	**60**	48	38		**74**	**73**	**70**	**76**	**73**	**77**	**72**	**72**	**74**	**61**	**65**	**62**
	RF	CART MSE	40	**75**	**57**	47	42	39	40	43	**56**	**56**	**63**	**61**	**64**	**65**	47	54	47	**57**	**58**	32	**71**	48	39	32	25	25	26		49	43	**56**	49	**57**	48	47	**53**	37	39	38
		CART RDE	40	**75**	**57**	47	43	39	40	43	**56**	**56**	**63**	**61**	**64**	**65**	48	55	47	**58**	**58**	32	**71**	49	40	32	25	25	27	**51**		44	**57**	49.8	**57**	49	47	**53**	38	40	38
		OC1	44	**79**	**60**	**52**	46	43	44	48	**62**	**61**	**67**	**67**	**69**	**70**	**53**	**60**	**53**	**62**	**61**	35	**75**	**53**	44	35	28	28	30	**57**	**56**		**63**	**56**	**63**	**55**	**55**	**59**	41	43	42
		PAIR MSE	37	**72**	**53**	43	39	36	37	40	**52**	**51**	**57**	**55**	**58**	**60**	42	**50.1**	41	**54**	**54**	29	**68**	44	35	28	22	22	24	44	43	37		43	**51**	42	41	47	34	37	35
		PAIR RDE	40	**76**	**57**	47	42	39	40	43	**57**	**56**	**63**	**62**	**65**	**66**	48	**55**	47	**58**	**58**	33	**72**	49	40	32	25	25	27	**51**	50.2	44	**57**		58	49	47	**53**	38	40	38
		SCRT MSE	37	**72**	**53**	43	39	36	37	39	**51**	**50.3**	**57**	**55**	**58**	**59**	42	49.7	41	**54**	**54**	30	**68**	44	34	28	22	22	23	43	43	37	49	42		41	40	46	34	36	35
		SCRT RDE	41	**76**	**58**	48	43	40	41	44	**58**	**57**	**64**	**63**	**65**	**66**	49	**56**	48	**59**	**59**	33	**73**	49.8	41	33	26	26	28	**52**	**51**	45	**58**	**51**	**59**		49.5	**55**	39	41	39
		SUPP MSE	43	**78**	**59**	49	44	41	42	46	**59**	**58**	**64**	**64**	**66**	**67**	49.7	**58**	49	**60**	**60**	34	**74**	**51**	41	33	26	26	28	**53**	**53**	45	**59**	**53**	**60**	**50.5**		**55**	40	42	40
		SUPP RDE	39	**74**	**55**	46	41	38	39	42	**54**	**53**	**61**	**59**	**61**	**62**	45	**52**	44	**56**	**57**	32	**70**	47	39	31	25	25	26	47	47	41	**53**	47	**54**	45	45		**37**	39	38
	lmm		49.97	**83**	**69**	**66**	**55**	**58**	**60**	**62**	**70**	**68**	**70**	**70**	**74**	**75**	**61**	**68**	**60**	**72**	**73**	41	**80**	**66**	**57**	47	37	36	39	**63**	**62**	**59**	**66**	**62**	**66**	**61**	**60**	**63**		**61**	**50.1**
	lq		49	**82**	**67**	**63**	**54**	**55**	**57**	**59**	**67**	**65**	**68**	**68**	**72**	**72**	**59**	**66**	**58**	**69**	**71**	40	**78**	**63**	**55**	45	33	33	35	**61**	**60**	**57**	**63**	**60**	**64**	**59**	**58**	**61**	39		45
knn	lmm		49.9	**83**	**67**	**66**	**56**	**59**	**60**	**63**	**70**	**68**	**70**	**70**	**74**	**75**	**61**	**68**	**60**	**71**	**72**	40	**79**	**63**	**56**	47	37	36	38	**62**	**62**	**58**	**65**	**62**	**65**	**61**	**60**	**62**	49.9	**55**	

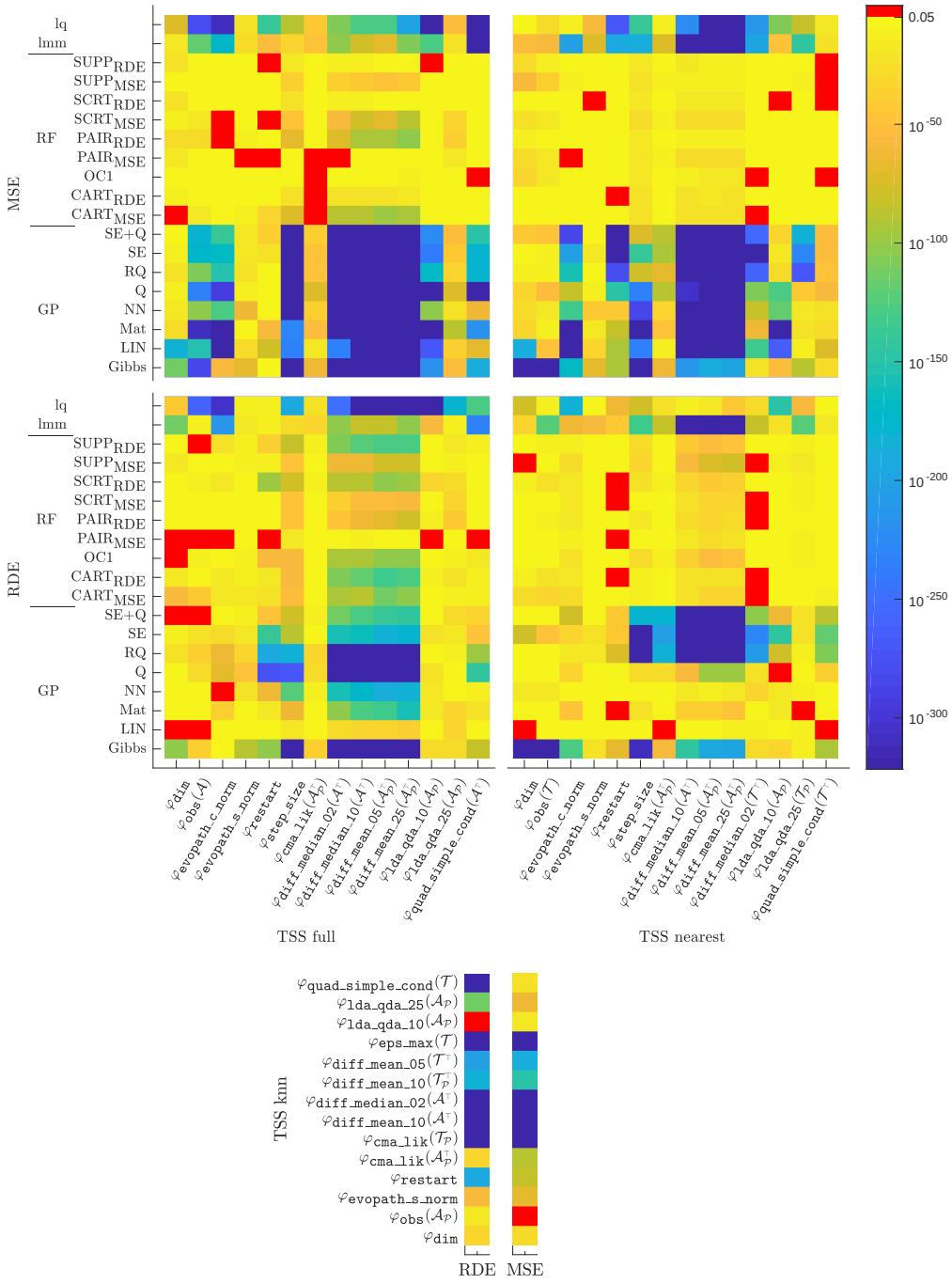


Figure 39: The visualisation of the p-values of the Kolmogorov-Smirnov test comparing the equality of probability distributions of individual features on all data and on those data on which a particular model setting scored best. Non-red colored squares denote KS test results rejecting the equality of both distributions with the Holm correction at the family-wise significance level $\alpha = 0.05$, otherwise, p-values are visualised as red squares. TSS knn was employed only in connection with lmm model.

From the model point of view, differences of RF model settings are much lower than the rest of settings for all the tested features. This might suggest the lower ability to distinguish between the individual RF settings maybe due to the low performance of these settings on the dataset. The p-values for the $\mathrm{err}_{\mathrm{MSE}}$ and the $\mathrm{err}_{\mathrm{RDE}}$ also differ notably mainly due to the different ranges of values of these two error measures. As its consequence, p-values for $\mathrm{err}_{\mathrm{MSE}}$ more clearly suggest the distinctive power of individual features on model precision, regardless the fact that $\mathrm{err}_{\mathrm{RDE}}$ is more convenient for CMA-ES related problems. Overall, the results of the KS test have shown that there is no tested landscape feature representative for which the selection of the covariance function would be negligible for the vast majority of model settings.

The most successful kind of models are GPs, present in most of the leaves proving the leading role shown in pairwise comparisons of model setting errors. On the other hand, Gibbs and Matérn $5/2$ covariances of the GP models constitute the winning GP settings regardless the fact that both provided the lowest numbers of predictions (see Table 43). This is probably caused by the removal of cases where predictions of any model setting were missing. The winning model setting of RF is OC1 method being the most often selected leaf of the classification tree for TSS full method. The polynomial models are not present in the resulting classification tree for TSS nearest. The feature $\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^{\top}_{\mathcal{P}})$ plays possibly the most important role in the classification tree for TSS full, being in the root node as well as in 4 other nodes, and also very important in the TSS nearest classification tree, where it is in the root node. This confirms the very strong decisive role of Φ_{Dis} features indicated in the results of KS test. Basic features $\varphi_{\texttt{dim}}$ and $\varphi_{\texttt{obs}}$ in few nodes provide the decisions resulting in GP model if both feature values are small and in RF model otherwise. Such dependency coincides with the better ability of RF models to preserve the prediction quality with the growing dimension than the ability of GP models. Features from the Φ_{Lvl} are also very important at least in connection with the TSS nearest. The Φ_{CMA} features are present only in very few nodes. We can observe the connection between the feature $\varphi^{\mathrm{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^{\top})$ and the lq model in the classification tree for TSS full showing the possible ability of Φ_{MM} features to detect convenience of polynomial model usage.

5.3.2 *Conclusion*

This section has presented an investigation of relationships between the prediction error of the surrogate models, their settings, and features describing the fitness landscape during evolutionary optimization by the CMA-ES state-of-the-art optimizer for expensive continuous black-box tasks. Four models in 39 different settings for the DTS-CMA-ES, were compared using three sets of 14 landscape features selected according to their properties. We analysed $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$ dependence of various models and model settings on the features calculated using three TSS extracted from DTS-CMA-ES runs on noiseless benchmarks from the COCO framework. Up to our knowledge, this analysis of landscape feature properties and their connection to surrogate models in evolutionary optimization is more profound than any other published so far.

The statistical analysis has shown significant differences in the error values among all 39 model settings using different TSS methods for both $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$. The GP model settings provided the highest perfomance followed by polynomial models. Most of the investigated features had distributions on sample sets with particular model settings achieving the

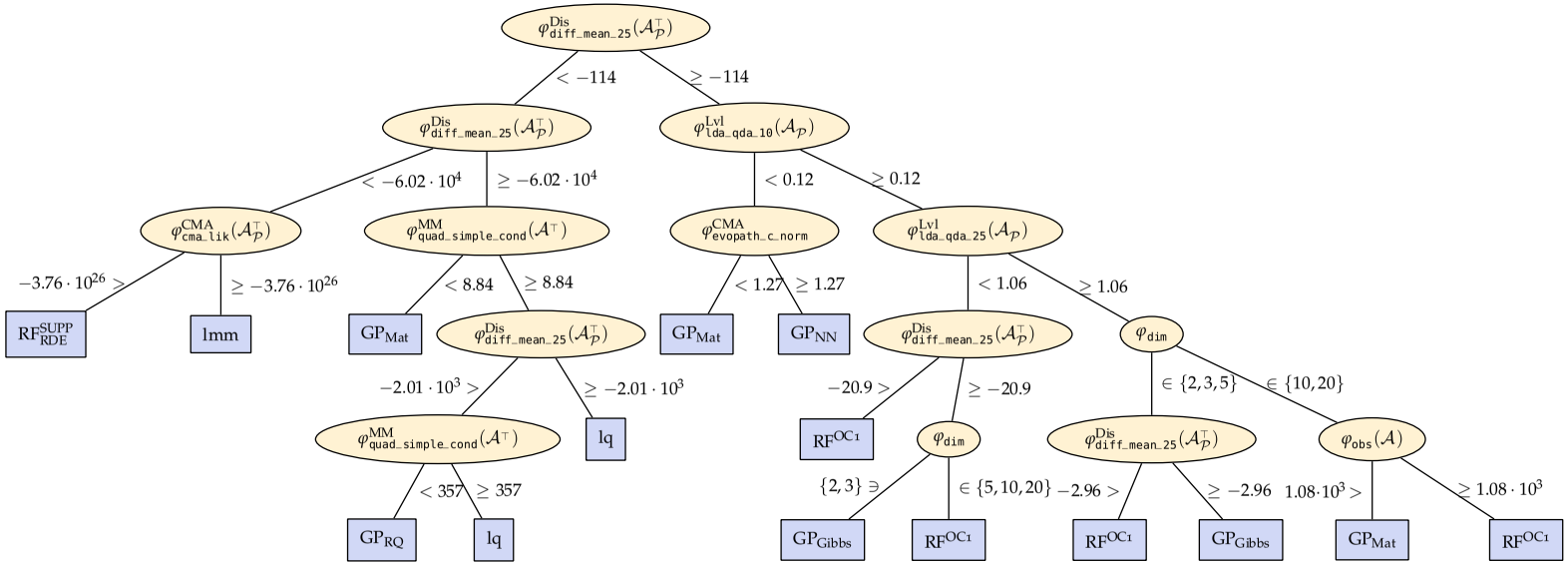


Figure 40: Classification tree analysing the influence of landscape features on the most suitable model and its setting using TSS full.

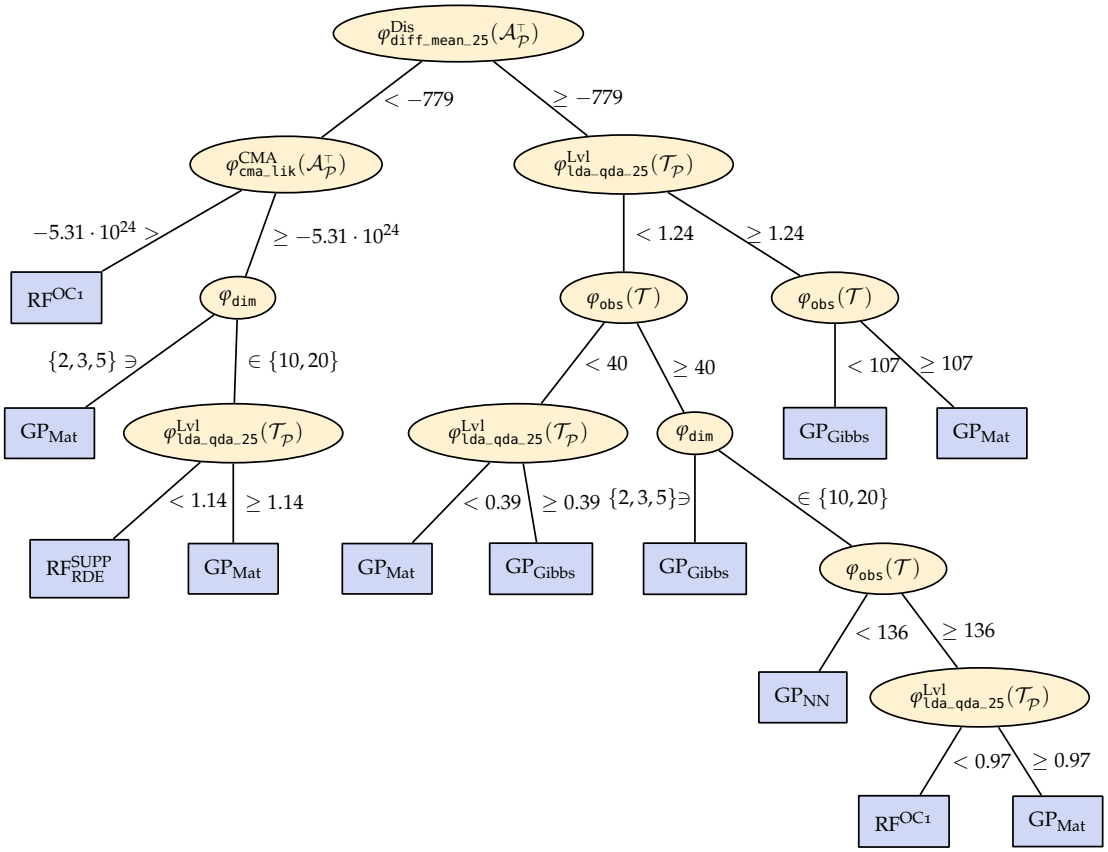


Figure 41: Classification tree analysing the influence of landscape features on the most suitable model and its setting using TSS nearest.

lowest error values significantly different from the overall distribution of all data for both error measures and all 3 TSS methods. Finally, the overall results have shown the dispersion features Φ_{Dis} as highly influential on the model settings error values, followed by features based on CMA-ES state variables Φ_{CMA}, features describing the similarity of fitness function to some simple model Φ_{MM}, features splitting space according to the black-box function values Φ_{Lvl}, and simple features such as dimension and number of observations.

Table 46: The p-values of the Kolmogorov-Smirnov (KS) test comparing the equality of probability distributions of individual TSS knn feature representatives on all data and on those data on which the lmm model setting scored best in $\mathrm{err}_{\mathrm{MSE}}$ and $\mathrm{err}_{\mathrm{RDE}}$. The p-values are after the Holm correction and they are shown only if the KS test rejects the equality of both distributions at the family-wise significance level $\alpha = 0.05$. Zeros indicate p-values below the smallest double precision number. SCRT = SECRET, SUPP = SUPPORT.

	$\varphi_{\texttt{dim}}$	$\varphi_{\texttt{obs}}(\mathcal{A}_\mathcal{P})$	$\varphi^{\mathrm{CMA}}_{\texttt{evopath\_s\_norm}}$	$\varphi^{\mathrm{CMA}}_{\texttt{restart}}$	$\varphi^{\mathrm{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top_\mathcal{P})$	$\varphi^{\mathrm{CMA}}_{\texttt{cma\_lik}}(\mathcal{T}_\mathcal{P})$	$\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{A}^\top)$	$\varphi^{\mathrm{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top)$	$\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_10}}(\mathcal{T}^\top_\mathcal{P})$	$\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{T}^\top)$	$\varphi^{\mathrm{Inf}}_{\texttt{eps\_max}}(\mathcal{T})$	$\varphi^{\mathrm{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_\mathcal{P})$	$\varphi^{\mathrm{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_\mathcal{P})$	$\varphi^{\mathrm{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{T})$
MSE	5.2e-32	9.2e-08	1.6e-63	1.9e-75	1.2e-78	3e-267	0	0	8.3e-126	7.4e-161	0	1.1e-23	2.8e-60	2e-25
RDE	9.5e-45	1.2e-28	2.8e-67	1.4e-196	1.6e-46	0	0	0	2.8e-188	4.9e-209	0	9.2e-14	7.4e-119	6.6e-319

Table 47: The p-values of the Kolmogorov-Smirnov (KS) test comparing the equality of probability distributions of individual TSS full feature representatives on all data and on those data on which a particular model setting scored best in $\mathrm{err}_{\mathrm{MSE}}$. The p-values are after the Holm correction and they are shown only if the KS test rejects the equality of both distributions at the family-wise significance level $\alpha = 0.05$, non-rejecting the equality hypothesis is indicated with —. Zeros indicate p-values below the smallest double precision number. SCRT = SECRET, SUPP = SUPPORT.

$\mathcal{M}$	GP								RF								lmm	lq
settings	Gibbs	LIN	Mat	NN	Q	RQ	SE	+Q	SEC MSE	CART RDE	CART MSE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE		
$\varphi_{\texttt{dim}}$	2e-101	—	3.2e-2	1.3e-3	2.9e-5	1.7e-23	2.8e-13	—	1.3e-59	1.9e-16	—	—	2.4e-2	3.9e-2	1.5e-3	1.0e-16	7.4e-3	4e-115	7.8e-46	
$\varphi_{\texttt{obs}}(\mathcal{A})$	1.8e-40	—	1.0e-7	1.7e-7	7.0e-29	5.9e-36	1.6e-21	—	2.6e-43	1.7e-7	8.4e-11	—	2.3e-10	1.1e-6	6.0e-15	1.4e-10	—	3.2e-14	3e-259	
$\varphi^{\mathrm{CMA}}_{\texttt{evopath\_c\_norm}}$	3.5e-7	3.5e-8	7.9e-37	—	2.6e-69	7.6e-62	9.1e-30	2.1e-10	9.9e-20	3.7e-18	4.5e-4	—	1.2e-8	9.3e-8	7.0e-18	2.2e-10	4.3e-24	1e-216	5e-307	
$\varphi^{\mathrm{CMA}}_{\texttt{evopath\_s\_norm}}$	1.3e-88	2.1e-3	9.7e-10	1.5e-25	3.3e-48	6.0e-33	9.2e-17	7.4e-15	5.1e-18	7.0e-13	6.9e-14	2.7e-6	2.6e-5	3.9e-5	1.3e-10	2.1e-14	6.3e-11	1.3e-16	1.1e-13	
$\varphi^{\mathrm{CMA}}_{\texttt{restart}}$	7e-105	1.5e-7	9.7e-20	2.3e-61	7e-273	5e-193	3e-137	3.5e-46	3.3e-21	8.2e-18	1.7e-60	—	2.0e-4	4.1e-9	8.7e-98	5.5e-14	5.1e-37	1.9e-12	2.2e-15	
$\varphi^{\mathrm{CMA}}_{\texttt{step\_size}}$	0	5.2e-19	1.5e-62	2e-123	1e-267	9e-183	6.1e-90	5.4e-79	5.0e-53	1.5e-57	1.1e-58	2.0e-20	3.4e-48	2.2e-47	2.5e-79	5.6e-46	8.4e-85	1.3e-28	9e-194	
$\varphi^{\mathrm{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^\top_\mathcal{P})$	1.5e-37	1.8e-2	8.2e-7	2.7e-14	2.9e-30	2.8e-33	3.8e-8	6.6e-14	1.7e-9	7.8e-9	4.3e-8	2.0e-3	2.2e-6	1.6e-5	9.3e-14	9.7e-8	4.5e-12	6.0e-34	1.5e-22	
$\varphi^{\mathrm{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^\top)$	0	2.7e-30	3e-103	3e-159	0	0	4e-160	2e-112	3.9e-94	7e-110	1.1e-91	6.1e-7	8.3e-50	2.3e-38	3.8e-89	5.9e-62	6e-107	3e-103	6e-250	
$\varphi^{\mathrm{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^\top)$	0	1.9e-35	1e-126	9e-173	0	0	3e-166	2e-130	2.8e-89	4e-118	5.0e-93	4.7e-11	1.8e-62	5.6e-53	2.4e-90	1.4e-61	4e-125	2.5e-82	0	
$\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^\top_\mathcal{P})$	0	1.6e-35	5e-132	7e-188	0	0	1e-185	7e-138	6e-115	2e-133	1e-104	3.1e-9	3.0e-68	5.8e-55	7e-102	6.5e-81	5e-129	2.0e-81	0	
$\varphi^{\mathrm{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^\top_\mathcal{P})$	0	6.2e-40	3e-157	6e-186	0	0	2e-181	4e-142	3e-102	8e-130	7e-102	3.9e-10	4.9e-77	7.2e-60	1.1e-99	1.7e-79	1e-128	1e-103	0	
$\varphi^{\mathrm{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_\mathcal{P})$	9.7e-29	1.7e-5	8.2e-18	3.4e-19	1.6e-22	4.4e-5	7.8e-25	1.5e-4	5.4e-4	5.4e-12	1.7e-10	—	1.0e-11	1.1e-7	1.7e-35	2.5e-3	1.7e-9	2.0e-56	0	
$\varphi^{\mathrm{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_\mathcal{P})$	6.9e-30	5.7e-33	6.2e-31	6.1e-8	3.8e-9	1.2e-12	2.6e-16	2.0e-29	7.2e-14	1.2e-23	3.1e-13	1.5e-6	1.6e-44	9.8e-33	3.9e-27	1.3e-14	1.2e-17	3.8e-14	3e-181	
$\varphi^{\mathrm{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^\top)$	7.6e-83	1.4e-9	3.5e-25	8.7e-13	2e-145	1.7e-96	2.9e-46	2.3e-34	8.3e-11	1.1e-11	3.9e-16	—	5.2e-5	5.5e-3	7.3e-8	3.0e-7	1.9e-13	7e-235	1e-127	

Table 48: The p-values of the Kolmogorov-Smirnov (KS) test comparing the equality of probability distributions of individual TSS full feature representatives on all data and on those data on which a particular model setting scored best in err_{RDE}. The p-values are after the Holm correction and they are shown only if the KS test rejects the equality of both distributions at the family-wise significance level $\alpha = 0.05$, non-rejecting the equality hypothesis is indicated with —. Zeros indicate p-values below the smallest double precision number. SCRT = SECRET, SUPP = SUPPORT.

$\mathcal{M}$	GP								RF										lmm	lq
settings	Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			
$\varphi_{\texttt{dim}}$	6e-112	3e-182	4.8e-30	4.8e-18	3.0e-14	1.6e-25	2.5e-14	2.2e-8	—	2.8e-12	1.2e-10	1.1e-17	1.1e-26	1.9e-15	1.5e-22	6.6e-5	1.3e-25	1.5e-17	4.1e-72	
$\varphi_{\texttt{obs}}(\mathcal{A})$	2e-286	2e-152	6e-308	5e-103	3e-232	4e-111	8e-175	7e-179	1.5e-17	2.5e-8	1.9e-2	1.4e-4	6.1e-26	4.0e-20	6.4e-5	5.5e-5	7.1e-6	4e-120	4e-280	
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	2.6e-56	0	0	3e-131	6e-295	7e-152	1e-170	7e-140	6.9e-4	9.5e-6	2.2e-3	1.2e-4	—	—	3.9e-6	1.4e-7	1.1e-3	1e-172	8e-309	
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	5.6e-69	8.8e-27	5.5e-4	3.1e-62	9.8e-13	6.1e-25	3.7e-16	9.0e-20	4.3e-11	7.3e-4	3.9e-2	—	1.4e-11	1.7e-5	1.3e-8	9.6e-4	3.3e-7	4.9e-27	1.7e-21	
$\varphi^{\text{CMA}}_{\texttt{restart}}$	4.3e-2	2.2e-77	1.7e-59	3.5e-6	3.1e-9	1.5e-7	2.2e-25	6.6e-37	5.1e-31	1.1e-4	3.8e-2	—	2.0e-2	—	3.3e-8	1.0e-4	—	1.7e-54	7e-138	
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	0	3e-234	1e-227	0	0	0	0	0	5.3e-75	2.3e-23	1.8e-4	5.5e-6	1.3e-68	2.7e-48	2.4e-23	5.0e-40	3.3e-17	8.7e-36	4.7e-87	
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^{\top}_{\mathcal{P}})$	6.3e-65	9.4e-19	1.3e-53	1.4e-36	2.1e-74	1.3e-48	1.3e-46	5.4e-38	—	—	—	—	7.8e-4	5.3e-3	4.2e-4	4.5e-2	5.8e-9	2.1e-46	5.1e-42	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{A}^{\top})$	0	9e-234	0	0	0	0	0	0	1.2e-88	1.7e-23	1.0e-7	—	2.7e-72	7.0e-57	9.6e-15	1.8e-37	3.3e-15	2e-104	5e-110	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^{\top})$	0	0	0	0	0	0	0	0	4.8e-88	3.6e-21	1.3e-9	7.6e-6	1.0e-93	3.3e-79	1.8e-13	1.1e-43	1.6e-12	3.8e-76	1e-130	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^{\top}_{\mathcal{P}})$	0	0	0	0	0	0	0	0	5.2e-98	4.2e-22	1.2e-9	8.6e-4	1.9e-93	4.7e-71	1.7e-8	3.6e-47	1.1e-8	2.9e-66	1e-128	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^{\top}_{\mathcal{P}})$	0	0	0	0	0	0	0	0	1.3e-93	1.3e-20	5.5e-11	6.1e-4	2e-103	2.8e-86	5.5e-9	2.0e-44	3.6e-8	1e-106	5e-201	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_{\mathcal{P}})$	6e-221	1e-268	0	1e-105	0	4e-167	2e-241	8e-223	6.8e-6	5.2e-4	1.7e-2	1.2e-7	3.5e-5	2.5e-9	5.1e-4	1.7e-2	—	4e-129	0	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{A}_{\mathcal{P}})$	1.6e-54	2.0e-40	3.4e-90	7.7e-24	1.3e-74	1.4e-35	6.3e-45	1.0e-51	5.2e-5	1.2e-3	5.2e-10	1.8e-18	3.0e-36	5.1e-33	3.3e-3	7.5e-5	1.1e-7	5.5e-25	4.8e-53	
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{A}^{\top})$	1e-190	8.5e-68	6e-220	2.1e-64	0	2e-177	3e-180	4e-148	7.1e-5	1.6e-3	—	8.8e-3	4.3e-9	6.0e-6	2.6e-4	2.0e-3	3.5e-2	0	0	

Table 49: The p-values of the Kolmogorov-Smirnov (KS) test comparing the equality of probability distributions of individual TSS nearest feature representatives on all data and on those data on which a particular model setting scored best in err_{MSE}. The p-values are after the Holm correction and they are shown only if the KS test rejects the equality of both distributions at the family-wise significance level $\alpha = 0.05$, non-rejecting the equality hypothesis is indicated with —. Zeros indicate p-values below the smallest double precision number. SCRT = SECRET, SUPP = SUPPORT.

$\mathcal{M}$	GP								RF										lmm	lq
settings	Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE			
$\varphi_{\texttt{dim}}$	0	—	1.9e-7	8.2e-19	3.8e-3	1.8e-9	5.5e-81	7.9e-3	9.7e-30	5.7e-14	2.9e-3	7.2e-3	2.9e-10	4.4e-7	2.5e-13	—	1.5e-2	6.7e-36	1.5e-77	
$\varphi_{\texttt{obs}}(\mathcal{T})$	0	5.0e-3	6.1e-6	6.0e-20	6.2e-6	2.2e-9	3.7e-50	1.5e-6	4.5e-35	4.7e-22	2.1e-9	2.0e-5	8.2e-13	2.2e-6	2.3e-24	9.3e-3	3.4e-11	2e-124	2.7e-12	
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	5e-131	6.1e-5	1.9e-46	1.1e-7	7.0e-32	5.2e-18	1.6e-33	1.5e-80	3.5e-26	7.9e-19	3.8e-23	5.5e-12	2.7e-19	2.2e-17	7.4e-19	4.7e-30	9.2e-13	2.6e-61	1e-183	
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	3.2e-25	2.1e-2	1.7e-3	7.9e-10	2.8e-4	2.5e-20	6.2e-21	1.0e-4	3.2e-9	1.1e-5	2.9e-7	1.4e-5	1.6e-7	8.8e-7	1.6e-6	4.6e-6	3.9e-5	1.7e-56	8.3e-7	
$\varphi^{\text{CMA}}_{\texttt{restart}}$	2.4e-69	2.5e-4	—	1.7e-2	1.2e-3	2.3e-75	3.8e-42	1.0e-45	2.6e-2	—	3.2e-2	—	3.3e-2	—	—	7.9e-4	2.5e-4	1.1e-42	7.0e-64	
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	8e-311	2.5e-5	2.4e-21	1.3e-4	1.6e-27	0	0	6e-178	6.3e-13	1.6e-14	6.6e-29	3.1e-8	2.1e-16	1.9e-9	9.7e-25	1.7e-34	6.3e-29	2e-136	2.0e-78	
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^{\top}_{\mathcal{P}})$	8.2e-60	—	6.1e-9	2.0e-4	1.1e-28	3e-182	1e-202	6e-180	1.0e-6	1.2e-4	1.2e-15	2.6e-7	1.9e-5	1.1e-6	1.4e-11	1.6e-10	2.8e-10	5.1e-83	3.7e-22	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^{\top})$	3e-146	1.9e-10	4.9e-37	7.4e-16	5.2e-57	0	0	0	3.8e-23	1.6e-22	3.8e-38	4.6e-18	5.0e-26	7.3e-23	1.3e-32	1.5e-60	5.9e-45	0	2.7e-14	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^{\top}_{\mathcal{P}})$	5e-192	2.4e-10	4.3e-38	4.6e-15	1e-100	0	0	0	6.3e-34	1.6e-25	9.3e-52	9.9e-27	5.7e-31	5.8e-35	5.2e-43	1.7e-73	2.2e-51	0	2.3e-19	
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^{\top}_{\mathcal{P}})$	4e-197	3.7e-11	6.3e-47	5.8e-18	8e-101	0	0	0	5.8e-31	2.3e-25	5.2e-50	9.3e-26	2.0e-30	5.4e-35	1.3e-40	1.0e-79	8.4e-50	0	9.6e-43	
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{T}^{\top})$	1.2e-26	1.9e-14	5.6e-7	2.0e-33	4.8e-21	2e-208	7e-228	3e-106	—	—	6.3e-5	2.5e-3	—	—	1.6e-4	—	5.0e-9	2.7e-81	2.1e-28	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_{\mathcal{P}})$	4.1e-35	2.2e-2	3.8e-16	1.4e-20	—	2.4e-76	8e-143	6.8e-66	1.8e-4	1.3e-5	3.9e-5	2.0e-7	3.4e-5	2.5e-6	1.2e-2	2.1e-14	6.2e-4	1.3e-21	4e-150	
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{T}_{\mathcal{P}})$	9.5e-15	3.2e-11	—	8.0e-14	2.9e-5	1.4e-15	5.7e-23	2.7e-22	1.1e-4	2.6e-5	6.1e-15	8.9e-15	1.6e-4	2.4e-12	9.7e-13	4.5e-16	3.7e-12	9e-123	9.8e-57	
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{T}^{\top})$	8.8e-96	—	6.8e-6	1.9e-10	3.3e-38	4e-111	2e-118	3.0e-90	2.2e-11	1.7e-8	5.2e-8	5.6e-4	2.7e-9	7.4e-3	9.1e-8	1.3e-4	4.4e-5	2.5e-20	3.6e-3	

Table 50: The p-values of the Kolmogorov-Smirnov (KS) test comparing the equality of probability distributions of individual TSS nearest feature representatives on all data and on those data on which a particular model setting scored best in err_{RDE}. The p-values are after the Holm correction and they are shown only if the KS test rejects the equality of both distributions at the family-wise significance level $\alpha = 0.05$, non-rejecting the equality hypothesis is indicated with —. Zeros indicate p-values below the smallest double precision number. SCRT = SECRET, SUPP = SUPPORT.

$\mathcal{M}$	GP								RF								lmm	lq
settings	Gibbs	LIN	Mat	NN	Q	RQ	SE+Q	SE	CART MSE	CART RDE	OC1	PAIR MSE	PAIR RDE	SCRT MSE	SCRT RDE	SUPP MSE	SUPP RDE		
$\varphi_{\texttt{dim}}$	0	2e-189	1.2e-31	1.5e-28	9.2e-32	2.6e-15	7.0e-10	5.6e-44	9.5e-7	2.3e-10	2.2e-34	2.4e-26	8.3e-11	1.5e-24	1.4e-8	4.3e-59	2.0e-29	7.8e-54	2.7e-7
$\varphi_{\texttt{obs}}(\mathcal{T})$	0	9.3e-50	1.2e-11	4.7e-16	7.0e-54	1.7e-15	1.5e-12	3.1e-48	2.1e-5	8.1e-5	7.5e-20	4.4e-16	7.3e-5	5.3e-16	9.8e-5	6.6e-37	1.0e-15	4.8e-58	1.2e-28
$\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}}$	2e-171	0	0	1e-132	6.0e-83	5e-159	4e-202	2e-284	3.3e-2	3.3e-2	2.2e-5	—	3.4e-2	1.5e-2	1.9e-3	4.2e-12	4.9e-5	3e-202	4.5e-13
$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}}$	1.0e-67	6.7e-34	1.0e-17	2.6e-45	4.1e-12	1.1e-9	5.8e-17	7.7e-10	5.3e-3	2.7e-3	1.4e-3	1.3e-3	2.1e-2	6.0e-6	—	7.3e-5	1.5e-4	2.2e-71	1.5e-59
$\varphi^{\text{CMA}}_{\texttt{restart}}$	4.0e-93	2.9e-91	1.9e-83	7.1e-43	1e-146	2e-280	0	2e-322	1.1e-2	—	1.6e-3	3.2e-4	3.5e-4	1.9e-4	4.4e-2	5.7e-7	3.4e-2	2e-196	3e-270
$\varphi^{\text{CMA}}_{\texttt{step\_size}}$	0	1e-236	0	2e-285	7e-234	8.6e-80	3e-139	2.8e-72	1.0e-22	5.3e-22	1.7e-19	3.5e-20	2.5e-17	9.8e-27	2.5e-16	3.9e-22	1.6e-20	2e-191	9e-138
$\varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathcal{A}^{\top}_{\mathcal{P}})$	6.9e-64	2.1e-14	1.4e-50	2.5e-31	2.5e-86	6.3e-71	5.2e-95	1.3e-86	4.3e-8	1.9e-8	9.0e-11	6.4e-11	2.1e-8	7.3e-13	1.0e-9	2.0e-11	3.4e-7	5.2e-72	3.4e-12
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_10}}(\mathcal{A}^{\top})$	1e-214	0	0	0	3e-307	0	0	0	1.2e-23	4.5e-24	1.5e-15	3.9e-27	1.2e-15	5.9e-28	3.8e-16	1.5e-15	4.6e-16	0	5e-225
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_05}}(\mathcal{A}^{\top}_{\mathcal{P}})$	1e-191	0	0	0	0	0	0	0	2.6e-22	8.3e-21	9.9e-20	1.6e-29	3.8e-15	1.9e-29	3.8e-19	9.5e-15	4.9e-14	0	0
$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_25}}(\mathcal{A}^{\top}_{\mathcal{P}})$	6e-197	0	0	0	0	0	0	0	2.6e-22	2.7e-21	1.7e-20	1.3e-30	9.8e-15	9.5e-31	3.6e-19	2.7e-15	5.0e-14	0	2e-317
$\varphi^{\text{Dis}}_{\texttt{diff\_median\_02}}(\mathcal{T}^{\top})$	1.0e-22	8e-132	9.6e-62	2.9e-84	1e-107	5e-262	0	4e-256	—	1.3e-3	—	1.2e-3	1.1e-3	5.3e-3	1.7e-2	3.4e-5	9.7e-6	6e-200	1e-215
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_10}}(\mathcal{A}_{\mathcal{P}})$	8.2e-52	3e-266	0	2e-116	9e-130	2.1e-26	1.4e-19	6.8e-31	7.1e-3	2.4e-5	3.5e-3	1.7e-5	8.8e-6	2.8e-4	—	1.7e-3	1.3e-2	1.8e-49	5e-107
$\varphi^{\text{Lvl}}_{\texttt{lda\_qda\_25}}(\mathcal{T}_{\mathcal{P}})$	6.8e-88	8.2e-12	1.0e-22	2.3e-17	1.2e-42	2e-268	9e-231	1e-186	4.0e-6	1.4e-3	5.1e-7	1.3e-10	6.1e-5	7.5e-10	2.7e-9	2.2e-11	2.5e-8	7e-146	8.2e-85
$\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{T}^{\top})$	3.4e-30	3.9e-71	4.2e-27	1.7e-22	3.3e-58	1.1e-46	9.8e-55	4.2e-55	2.4e-4	6.2e-4	—	1.2e-3	3.5e-3	6.8e-4	—	—	—	1.1e-21	6.0e-16

6 CONCLUSIONS

In this thesis, we have explored many approaches to surrogate modeling in continuous evolutionary optimization of expensive black-box problems. What have we learned about surrogate-assisted optimization, and what are the most promising direction for future research? This final chapter seeks to answer these questions.

6.1 SUMMARY OF CONTRIBUTIONS

Contributions in this thesis were structured into two parts in order to represent two aspects of my research into surrogate-assisted evolutionary black-box optimization that contributed to the stated goals: *Designing a version of the* CMA-ES assisted by a surrogate model capable to predict the entire distribution of the optimized fitness and *Investigating the features describing the fitness landscape and their relationships to surrogate models*. As described within this work, we have made valuable progress in each of these categories with the most important ones among them being summarized below.

SURROGATE MODELING FOR THE CMA-ES In Chapter 4, we have first shown that the approach based on Gaussian processes and random forests can be efficiently used to build surrogates models for surrogate-assisted CMA-ES in the black-box scenario, when the landscape of the objective function is unknown. We have used the two models in combination with the CMA-ES in two evolution control schemes: generation and doubly trained evolution controls resulting in the S-CMA-ES and DTS-CMA-ES algorithms. The S-CMA-ES mostly outperform the CMA-ES in initial phases of the optimization process. For its adaptive version, we have found two error measures, the Kendall rank correlation and the ranking difference error, that significantly outperformed the non-adptive version used with a higher number of model-evaluated generations, especially in higher dimensionalities of the input space. Our doubly trained evolution control utilized in the DTS-CMA-ES improved the performance of the original algorithm even more by using the ability of Gaussian processes and random forests to predict the entire distribution of the optimized function. The DTS-CMA-ES variant with GPs represents an algorithm of choice for multimodal functions with weak global structure and is very eligible for unimodal landscapes, too, especially in lower dimensions. Its self-adaptive version, on the other hand, excels on the globally decreasing multimodal functions where it outperforms wast majority of surrogate-assisted algorithms. In our research into the connection between the surrogate model and its control in the surrogate-assisted evolutionary optimizer, we have found significant differences as to the performance of different evolution controls, models, and their combinations. The important finding was that both the model and the evolution control has significant influence on the convergence speed.

Throughout the Chapter 4, we have provided multiple evaluations of different parameter settings for GP models. The selection of the k nearest neighbor points as training sets in a specified range for surrogates performed best. We have also shown statistical equivalence of rational quadratic, squared exponential, and Matérn $5/2$, in the sense that for no function, no

group of functions, no dimension, and no function-dimension combination, none of these covariance functions was significantly better than the other on the data from the noiseless part of the COCO platform. We have compared the ordinal GP regression model with the metric GP regression model used in the DTS-CMA-ES. The performance of the ordinal models was considerably lower than the standard GP models with few exceptions (e. g., the Attractive sector function). Up to our knowledge, we have also presented the first comparison of GP covariance functions in combination with ANN in the context of optimization data. Considering automated selection of a GP model, our presented algorithm using Bayesian model comparison techniques tested only on preliminary experiments has shown the need for more sophisticated covariance functions. However, due to the small number of experiments performed so far, it is difficult to draw any serious conclusions. Gaussian processes outperformed random forests both with respect to predictive accuracy and in using the models in connection with the CMA-ES. On the other hand, we found that also the RF model usually reduces the number of fitness evaluations required by the CMA-ES, especially on multi-modal functions. The split algorithms for decision trees in RF based on the classification of the input points provide slightly better performance as to the CMA-ES convergence than the other algorithms tested.

LANDSCAPE ANALYSIS FOR SURROGATE MODELING In Chapter 5, we have investigated the landscape features and surrogate models in the context of evolutionary black-box optimizer in two different scenarios. In our research into the static scenario, utilized e. g., in (Saini et al., 2019), where the model is selected once for a specific problem at the beginning of the optimization process, we have shown that the features describing global properties of the landscape are highly influential on surrogate model settings performance. On the other hand, the research have also suggested the difficulties of derivation of clear relationships between the performance of the compared settings of GP and RF models and the considered features. This lead us to more intensive investigation in the properties of the landscape features. We have selected three sets of 14 landscape features as valuable characteristics of the fitness landscape according to their properties, especially according to their robustness and similarity to other features. Up to our knowledge, this analysis of landscape feature properties in connection with surrogate models in evolutionary optimization is first of its kind especially in the dynamic scenario, where we consider the changes of the model performance between generations as the CMA-ES searches the optimimum of the investigated landscape. Our contribution to landscape features was also a set of new features based on the CMA-ES state variables. In the dynamic scenario, we have also shown significant differences in the error values among 39 model settings using different methods for selection of traning point on two error measures. Most of the investigated features had distributions on sample sets with particular model settings achieving the lowest error values significantly different from the overall distribution of all data for both error measures and all 3 TSS methods. The overall results have shown the dispersion features as highly influential on the model settings error values as well as plenty of other features including even simple features such as dimension and number of observations.

6.2 FUTURE DIRECTIONS

According to the experimental results, different surrogate CMA-ES algorithms or their variants perform best on different fitness landscapes, but the mechanism of selecting the best-

performing setting or smarter adaptation to the fitness at hand is still an open issue. Such adaptivity should select the appropriate evolution control parameters as well as the most convenient surrogate model and its parameters, and, in particular, it should decide whether to use the metric-based (probably Gaussian processes) or the ranking-based (e. g., ranking SVR) model. Besides, the parameters of the CMA-ES itself are problem-dependent, too. In fact, CMA-ES variants more suitable for expensive problems have been already proposed (Belkhir et al., 2017; Loshchilov, 2017), and they can be obviously combined with surrogate models.

The research supporting such adaptivity considering surrogate models and their settings is part of this thesis. The issues of the first attempts in Section 4.6 should be fixed the research extended, e. g., into a combinatorial search over kernels in flavor of (Duvenaud et al., 2013; Gagné et al., 2006), a *co-evolution* of an ensemble of covariance functions alongside the population of candidate solutions to the black-box objective function, or application of surrogate modeling to high-dimensional problems using algorithms for variable selection via multiple kernel learning (Bach, 2009; Duvenaud et al., 2011).

The results of our research into relationships of landscape features and surrogate models could definitely be utilized to design a metalearning-based system (Kerschke et al., 2019; Saini et al., 2019) for selection of surrogate models in surrogate-assisted evolutionary optimization algorithms context. Although some initial steps towards such system have already been made (see (Dvořák et al., 2020; Pitra et al., 2019a)), this research field is one of the most important ones to address in the near future.

Considering the results of the connection of the GP and ANN, the reconsideration of the approach employed in GPyTorch – training the ANN and the GP forming the combined surrogate model together by means of likelihood maximization — is on hand. Whereas maximum likelihood is indeed the commonly used objective function for GP learning (Rasmussen and Williams, 2006), successful and efficient ANN learning algorithms typically rely on other objectives (Goodfellow et al., 2016). Therefore, the investigation of a cyclic interleaving of a GP-learning phase with an ANN-learning phase, where the length of the latter will depend on the relationship of its success to the success of the former, could bring interesting results. As to the metalearning-based system, an ANN-GP model with a deep neural network could be trained on data from many optimization runs, and then the model used in a new run of the same optimizer may possibly be obtained through additional learning restricted only to the GP and the last 1-2 layers of the ANN.

BIBLIOGRAPHY

N. A. Ahmed and D. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35(3):688–692, 1989. doi: 10.1109/18.30996. URL https://doi.org/10.1109/18.30996.

H. Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, 1973.

C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11 (1-5):11–73, 1997. doi: 10.1023/A:1006559212014.

A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *The 2005 IEEE Congress on Evolutionary Computation, 2005*, volume 2, pages 1769–1776. IEEE, 2005. doi: 10.1109/CEC.2005.1554902.

A. Auger, M. Schoenauer, and N. Vanhaecke. LS-CMA-ES: A Second-Order Algorithm for Covariance Matrix Adaptation. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 182–191. Springer Berlin Heidelberg, 2004.

A. Auger, D. Brockhoff, and N. Hansen. Benchmarking the local metamodel CMA-ES on the noiseless BBOB'2013 test bed. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1225–1232, New York, NY, USA, 2013. ACM. doi: 10.1145/2464576.2482701.

F. Bach. High-dimensional non-linear variable selection through hierarchical kernel learning, 2009.

M. Baerns and M. Holeňa. *Combinatorial Development of Solid Catalytic Materials. Design of High-Throughput Experiments, Data Analysis, Data Mining*. Imperial College Press / World Scientific, London, 2009.

L. Bajer and M. Holeňa. *Surrogate Model for Continuous and Discrete Genetic Optimization Based on RBF Networks*, pages 251–258. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi: 10.1007/978-3-642-15381-5_31. URL http://dx.doi.org/10.1007/978-3-642-15381-5_31.

L. Bajer. *Model-based evolutionary optimization methods*. PhD thesis, Faculty of Mathematics and Physics, Charles University in Prague, Prague, 2018.

L. Bajer and Z. Pitra. Surrogate CMA-ES. https://github.com/bajeluk/surrogate-cmaes, 2014.

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, pages 1143–1150, New York, NY, USA, 2015. Association for Computing Machinery. doi: 10.1145/2739482.2768468. URL https://doi.org/10.1145/2739482.2768468.

L. Bajer, Z. Pitra, J. Repický, and M. Holeňa. Gaussian process surrogate models for the CMA Evolution Strategy. *Evolutionary Computation*, 27(4):665–697, 2019. doi: 10.1162/evco_a_00244. URL https://doi.org/10.1162/evco_a_00244.

P. Baudiš and P. Pošík. Global line search algorithm hybridized with quadratic interpolation and its extension to separable functions. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 257–264, New York, NY, USA, 2015. ACM. doi: 10.1145/2739480.2754717. URL http://doi.acm.org/10.1145/2739480.2754717.

BBOB. The Black-Box Optimization Benchmarking (BBOB) workshop series, 2009–2022. URL {https}://numbbo.github.io/workshops/.

B. K. Beachkofski and R. V. Grandhi. Improved distributed hypercube sampling. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA paper 2002-1274. Americal Institute of Aeronautics and Astronautics, 2002.

J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. Van der Meulen. Nonparametric entropy estimation : an overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

A. Bekasiewicz and S. Koziel. Surrogate-assisted design optimization of photonic directional couplers. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 30(3-4), 2017.

N. Belkhir, J. Dréo, P. Savéant, and M. Schoenauer. Per instance algorithm configuration of CMA-ES with limited budget. In P. A. N. Bosman, editor, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017*, pages 681–688. ACM, 2017. doi: 10.1145/3071178.3071343. URL https://doi.org/10.1145/3071178.3071343.

Y. Bengio and Y. Lecun. *Scaling learning algorithms towards AI*. MIT Press, 2007.

B. Bischl, O. Mersmann, H. Trautmann, and M. Preuß. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In T. Soule and J. H. Moore, editors, *GECCO*, pages 313–320. ACM, 2012.

Z. Bouzarkouna, A. Auger, and D. Y. Ding. Investigating the Local-Meta-Model CMA-ES for Large Population Sizes. In C. D. Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcázar, C. K. Goh, and J. J, editors, *3rd European event on Bio-inspired algorithms for continuous parameter optimisation (EvoNUM'10)*, volume 6024 of *Lecture Notes in Computer Science*, pages 402–411, Istanbul, Turkey, 2010.

Z. Bouzarkouna, A. Auger, and D. Y. Ding. Local-meta-model CMA-ES for partially separable functions. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 869–876, New York, NY, USA, 2011. ACM. doi: 10.1145/2001576.2001695.

L. Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman. Arcing classifier. *The Annals of Statistics*, 26(3):801–849, 1998.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

R. P. Brent. *Algorithms for Minimization Without Derivatives*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, New Jersey, 1973.

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

T. D. Bui, D. Hernández-Lobato, J. M. Hernández-Lobato, Y. Li, and R. E. Turner. Deep Gaussian processes for regression using approximate expectation propagation. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1472–1481. JMLR.org, 2016.

R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

R. Calandra, J. Peters, C. Rasmussen, and M. Deisenroth. Manifold Gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3338–3345. IEEE, 2016.

P. Chaudhuri, M.-C. Huang, W.-Y. Loh, and R. Yao. Piecewise-polynomial regression trees. *Statistica Sinica*, 4(1):143–167, 1994.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, KDD '16, pages 785–794. ACM, 2016.

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *J. Mach. Learn. Res.*, 6:1019–1041, 2005.

T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9):3137–3166, 2019. doi: 10.1007/s00500-017-2965-0. URL https://doi.org/10.1007/s00500-017-2965-0.

N. L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proc. of the International Conference on Genetic Algorithms and Their Applications*, pages 183–187, Pittsburgh, PA, 1985.

A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research, 2011.

K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep Gaussian processes. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR, 2017.

C. Darwin. *On the origin of species*. D. Appleton and Co., New York, 1859.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

B. Derbel, A. Liefooghe, S. Verel, H. E. Aguirre, and K. Tanaka. New features for continuous exploratory landscape analysis based on the SOO tree. In T. Friedrich, C. Doerr, and D. V. Arnold, editors, *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, FOGA 2019, Potsdam, Germany, August 27-29, 2019*, FOGA '19, pages 72–86. ACM, 2019. doi: 10.1145/3299904.3340308. URL https://doi.org/10.1145/3299904.3340308.

A. Dobra and J. Gehrke. SECRET: A scalable linear regression tree algorithm. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, KDD '02, pages 481–487. ACM, 2002. doi: 10.1145/775047.775117. URL https://doi.org/10.1145/775047.775117.

D. Duvenaud, H. Nickisch, and C. E. Rasmussen. Additive gaussian processes. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 226–234, Red Hook, NY, USA, 2011. Curran Associates Inc.

D. Duvenaud, J. Lloyd, R. Grosse, J. Tenenbaum, and G. Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1166–1174, Atlanta, Georgia, USA, 2013. PMLR. URL http://proceedings.mlr.press/v28/duvenaud13.html.

D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

M. Dvořák, Z. Pitra, and M. Holeňa. Assessment of surrogate model settings using landscape analysis. In M. Holeňa, T. Horváth, A. Kelemenová, F. Mráz, D. Pardubská, M. Plátek, and P. Sosík, editors, *Proceedings of the 20th Conference Information Technologies - Applications and Theory (ITAT 2020), Hotel Tyrapol, Oravská Lesná, Slovakia, September 18-22, 2020*, volume 2718 of *CEUR Workshop Proceedings*, pages 81–89. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2718/paper20.pdf.

A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. *Metamodel—Assisted Evolution Strategies*, pages 361–370. Springer Berlin Heidelberg, 2002.

M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. URL http://coco.gforge.inria.fr/bbob2010-downloads. Updated February 2010.

C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. Barrier Trees of Degenerate Landscapes. *Zeitschrift für Physikalische Chemie International Journal of Research in Physical Chemistry and Chemical Physics*, 216(2):155–173, 2002.

L. Fogel, A. Owens, and M. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, 1966.

A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, Ltd, 2008.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29 (5):1189–1232, 2001.

J. H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, 2002.

C. Gagné, M. Schoenauer, M. Sebag, and M. Tomassini. Genetic programming for kernel-based learning with co-evolving subsets selection. In T. P. Runarsson, H. Beyer, E. K. Burke, J. J. M. Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference, Reykjavik, Iceland, September 9-13, 2006, Procedings*, volume 4193 of *Lecture Notes in Computer Science*, pages 1008–1017. Springer, 2006. doi: 10.1007/11844297_102. URL https://doi.org/10.1007/11844297_102.

A. Galántai. The theory of Newton's method. *Journal of Computational and Applied Mathematics*, 124(1–2): 25–44, 2000. doi: http://doi.org/10.1016/S0377-0427(00)00435-0. URL http://www.sciencedirect.com/science/article/pii/S0377042700004350. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

S. García and F. Herrera. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration, 2019.

A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014. doi: 10.1007/s11222-013-9416-2. URL http://dx.doi.org/10.1007/s11222-013-9416-2.

M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Department of Physics, University of Cambridge, 1997.

I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. URL http://www.deeplearningbook.org/.

H. Haario, M. Laine, A. Mira, and E. Saksman. Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16(4):339–354, 2006.

N. Hansen. A global surrogate assisted CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pages 664–672, New York, NY, USA, 2019. ACM.

N. Hansen, S. Finck, R. Ros, and A. Auger. Real-Parameter Black-Box Optimization Benchmarking 2009: Noisy Functions Definitions. Research Report RR-6869, INRIA, 2009a.

N. Hansen. The CMA Evolution Strategy: A Comparing Review. In J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors, *Towards a New Evolutionary Computation*, number 192 in Studies in Fuzziness and Soft Computing, pages 75–102. Springer Berlin Heidelberg, 2006.

N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 Function Testbed. In *Proceedings of the 11th Annual GECCO Conference Companion: Late Breaking Papers*, GECCO '09, pages 2389–2396, New York, NY, USA, 2009a. ACM. doi: 10.1145/1570256.1570333.

N. Hansen. Benchmarking the Nelder-Mead Downhill Simplex Algorithm With Many Local Restarts. In *ACM-GECCO Genetic and Evolutionary Computation Conference*, Montreal, Canada, 2009b.

N. Hansen. Injecting external solutions into CMA-ES, 2011.

N. Hansen and A. Auger. Principled design of continuous stochastic search: From theory to practice. In *Theory and Principled Methods for the Design of Metaheuristics*, pages 145–180. Springer Berlin Heidelberg, 2014.

N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation, 1996.*, pages 312–317. IEEE, 1996.

N. Hansen and R. Ros. Benchmarking a weighted negative covariance matrix update on the BBOB-2010 noiseless testbed. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1673–1680. ACM, 2010.

N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009b. Updated February 2010.

N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012. URL http://coco.lri.fr/BBOB-downloads/download11.05/bbobdocexperiment.pdf.

N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *CoRR*, abs/1603.08785, 2016. URL http://arxiv.org/abs/1603.08785.

N. Hansen, A. Auger, R. Ros, O. Merseman, T. Tušar, and D. Brockhoff. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021. doi: 10.1080/10556788.2020.1808977. URL https://doi.org/10.1080/10556788.2020.1808977.

A. Hebbal, L. Brevault, M. Balesdent, E. Taibi, and N. Melab. Efficient global optimization using deep Gaussian processes. In *2018 IEEE Congress on Evolutionary Computation, CEC 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–8. IEEE, 2018.

R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99.*, volume 1, pages 97–102, 1999.

G. Hernández-Muñoz, C. Villacampa-Calvo, and D. Hernández-Lobato. Deep Gaussian processes using expectation propagation and Monte Carlo methods. In F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part III*, volume 12459 of *Lecture Notes in Computer Science*, pages 479–494. Springer, 2020.

G. E. Hinton and M. Revow. Using pairs of data-points to define splits for decision trees. In *Advances in Neural Information Processing Systems*, volume 8, pages 507–513. MIT Press, 1996.

M. Holeňa, D. Linke, and L. Bajer. Surrogate modeling in the evolutionary optimization of catalytic materials. In T. Soule, editor, *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 1095–1102, New York, NY, USA, 2012. ACM. doi: 10.1145/2330163.2330315. URL http://doi.acm.org/10.1145/2330163.2330315.

J. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* University of Michigan Press, 1975.

H. H. Hoos, T. Peitl, F. Slivovsky, and S. Szeider. Portfolio-based algorithm selection for circuit QBFs. In J. N. Hooker, editor, *Principles and Practice of Constraint Programming - 24th International Conference, CP 2018, Lille, France, August 27-31, 2018, Proceedings*, volume 11008 of *Lecture Notes in Computer Science*, pages 195–209. Springer, 2018.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pages 507–523, Berlin, Heidelberg, 2011. Springer-Verlag. doi: 10.1007/978-3-642-25566-3_40.

F. Hutter, H. Hoos, and K. Leyton-Brown. An Evaluation of Sequential Model-based Optimization for Expensive Blackbox Functions. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1209–1216, New York, NY, USA, 2013. ACM. doi: 10.1145/2464576.2501592.

G. A. Jastrebski and D. V. Arnold. Improving Evolution Strategies through Active Covariance Matrix Adaptation. In *IEEE Congress on Evolutionary Computation – CEC 2006*, pages 2814–2821, 2006. doi: 10.1109/cec.2006.1688662.

Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9 (1):3–12, 2005. doi: 10.1007/s00500-003-0328-5.

Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1:61–70, 2011.

Y. Jin, M. Hüsken, M. Olhofer, and B. Sendhoff. *Neural Networks for Fitness Approximation in Evolutionary Optimization*, pages 281–306. Springer Berlin Heidelberg, 2005.

Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 1, pages 592–599, 2001.

Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, 2002. doi: 10.1109/TEVC.2002.800884.

D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, 1998. doi: 10.1023/A:1008306431147.

S. Kern, N. Hansen, and P. Koumoutsakos. Local Meta-models for Optimization Using Evolution Strategies. In *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 939–948. Springer Berlin Heidelberg, 2006.

P. Kerschke. *Automated and Feature-Based Problem Characterization and Algorithm Selection Through Machine Learning*. Dissertation, University of Münster, 2017a. URL http://nbn-resolving.de/urn:nbn:de:hbz:6-39169612241. Publication status: Published.

P. Kerschke. Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package flacco. *ArXiv e-prints*, 2017b.

P. Kerschke and J. Dagefoerde. *flacco: Feature-Based Landscape Analysis of Continuous and Constraint Optimization Problems*, 2017. URL https://cran.r-project.org/package=flacco. R-package v. 1.7.

P. Kerschke, M. Preuss, C. Hernández, O. Schütze, J.-Q. Sun, C. Grimme, G. Rudolph, B. Bischl, and H. Trautmann. *Cell Mapping Techniques for Exploratory Landscape Analysis*, pages 115–131. Springer International Publishing, Cham, 2014. doi: 10.1007/978-3-319-07494-8_9.

P. Kerschke, M. Preuss, S. Wessing, and H. Trautmann. Detecting funnel structures by means of exploratory landscape analysis. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 265–272, New York, NY, USA, 2015. ACM. doi: 10.1145/2739480.2754642.

P. Kerschke, M. Preuss, S. Wessing, and H. Trautmann. Low-budget exploratory landscape analysis on multiple peaks models. GECCO '16, pages 229–236. ACM, 2016.

P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation*, 27(1):3–45, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

J. Koza, J. Tumpach, Z. Pitra, and M. Holeňa. Combining Gaussian processes and neural networks in surrogate modelling for covariance matrix adaptation evolution strategy. In B. Brejová, L. Ciencialová, M. Holeňa, F. Mráz, D. Pardubská, M. Plátek, and T. Vinař, editors, *Proceedings of the 21st Conference Information Technologies - Applications and Theory (ITAT 2021), Hotel Heľpa, Nízke Tatry and Muránska planina, Slovakia, September 24-28, 2021*, volume 2962 of *CEUR Workshop Proceedings*, pages 29–38. CEUR-WS.org, 2021a. URL http://ceur-ws.org/Vol-2962/paper27.pdf.

J. Koza, J. Tumpach, Z. Pitra, and M. Holeňa. Using past experience for configuration of gaussian processes in black-box optimization. In D. E. Simos, P. M. Pardalos, and I. S. Kotsireas, editors, *Learning and Intelligent Optimization - 15th International Conference, LION 15, Athens, Greece, June 20-25, 2021, Revised Selected Papers*, volume 12931 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2021b. doi: 10.1007/978-3-030-92121-7_15. URL https://doi.org/10.1007/978-3-030-92121-7_15.

G. Kronberger and M. Kommenda. Evolution of covariance functions for Gaussian process regression using genetic programming. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, editors, *Computer Aided Systems Theory - EUROCAST 2013 - 14th International Conference, Las Palmas de Gran Canaria, Spain, February 10-15, 2013, Revised Selected Papers, Part I*, volume 8111 of *Lecture Notes in Computer Science*, pages 308–315. Springer, 2013. doi: 10.1007/978-3-642-53856-8_39. URL https://doi.org/10.1007/978-3-642-53856-8_39.

J. Kruisselbrink, M. Emmerich, A. Deutz, and T. Bäck. A robust optimization approach using Kriging metamodels for robustness approximation in the CMA-ES. In *2010 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

H. Lee, Y. Jo, D. Lee, and S. Choi. Surrogate model based design optimization of multiple wing sails considering flow interaction effect. *Ocean Engineering*, 121:422–436, 2016.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, pages 1–17. OpenReview.net, 2018.

K.-H. Liang, X. Yao, and C. Newton. Evolutionary search of approximated n-dimensional landscapes. *International Journal of Knowledge-based Intelligent Engineering Systems*, 4:172–183, 2000.

J. R. Lloyd, D. Duvenaud, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani. Automatic construction and natural-language description of nonparametric regression models. *CoRR*, abs/1402.4304, 2014. URL http://arxiv.org/abs/1402.4304.

I. Loshchilov. LM-CMA: an alternative to L-BFGS for large-scale black box optimization. *Evolutionary Computation*, 25(1):143–171, 2017. doi: 10.1162/EVCO_a_00168. URL https://doi.org/10.1162/EVCO_a_00168.

I. Loshchilov, M. Schoenauer, and M. Sebag. Comparison-Based Optimizers Need Comparison-Based Surrogates. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 364–373. Springer Berlin Heidelberg, 2010.

I. Loshchilov, M. Schoenauer, and M. Sebag. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In *Proceedings of the 14th GECCO*, GECCO '12, pages 321–328, New York, NY, USA, 2012. ACM.

I. Loshchilov, M. Schoenauer, and M. Sebag. BI-population CMA-ES Algorithms with Surrogate Models and Line Searches. In *Genetic and Evolutionary Computation Conference (GECCO Companion)*, pages 1177–1184. ACM Press, 2013a.

I. Loshchilov, M. Schoenauer, and M. Sebag. Intensive surrogate model exploitation in self-adaptive surrogate-assisted CMA-ES (saACM-ES). In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 439–446. ACM Press, 2013b.

I. Loshchilov, M. Schoenauer, and M. Sebag. KL-based control of the learning schedule for surrogate black-box optimization. *CoRR*, abs/1308.2655, 2013c. URL http://arxiv.org/abs/1308.2655.

J. Lu, B. Li, and Y. Jin. An evolution strategy assisted by an ensemble of local gaussian process models. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 447–454, New York, NY, USA, 2013. ACM. doi: 10.1145/2463372.2463425.

M. A. Luersen and R. Le Riche. Globalized Nelder–Mead method for engineering optimization. *Computers & Structures*, 82(23):2251–2260, 2004.

M. Lunacek and D. Whitley. The dispersion metric and the CMA evolution strategy. GECCO '06, pages 477–484. ACM, 2006.

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley and Sons, Chichester, 2007.

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, pages 1–15. OpenReview.net, 2018.

O. Mersmann, M. Preuss, and H. Trautmann. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. PPSN XI, pages 73–82. Springer Berlin Heidelberg, 2010.

O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 829–836, New York, NY, USA, 2011. ACM. doi: 10.1145/2001576.2001690.

J. Mockus, V. Tiesis, and A. Zilinskas. The application of bayesian methods for seeking the extremum. *Toward Global Optimization*, 2:117–129, 1978.

H. Mohammadi, R. Riche, and E. Touboul. Making EGO and CMA-ES Complementary for Global Optimization. In C. Dhaenens, L. Jourdan, and M.-E. Marmion, editors, *Learning and Intelligent Optimization*, volume 8994 of *Lecture Notes in Computer Science*, pages 287–292. Springer International Publishing, 2015.

M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge. Algorithm selection for black-box continuous optimization problems. *Inf. Sci.*, 317(C):224–245, 2015.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *J. Artif. Int. Res.*, 2(1):1–32, 1994.

M. A. Muñoz, M. Kirley, and S. K. Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Transactions on Evolutionary Computation*, 19(1): 74–87, 2015.

R. Myers and D. Montgomery. *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1995.

L. Na, Q. Feng, Z. Liang, and W.-m. Zhong. Gaussian process assisted coevolutionary estimation of distribution algorithm for computationally expensive problems. *Journal of Central South University of Technology*, 19:443–452, 2012. doi: 10.1007/s11771-012-1023-4.

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4): 308–313, 1965.

R. Novak, L. Xiao, Y. Bahri, J. Lee, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, pages 1–35. OpenReview.net, 2019.

S. Nowozin. Improved information gain estimates for decision tree induction. *CoRR*, abs/1206.4620, 2012.

Y. S. Ong, P. B. Nair, and A. J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4):687–696, 2003.

Y. Ong, P. Nair, A. Keane, and K. Wong. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 307–331. Springer, 2005.

Z. Pitra, L. Bajer, and M. Holeňa. Comparing SVM, Gaussian Process and Random Forest Surrogate Models for the CMA-ES. In J. Yaghob, editor, *ITAT 2015: Information Technologies - Applications and Theory*, volume 1422, pages 186–193, North Charleston, USA, 2015. CreateSpace Independent Publishing Platform. URL http://ceur-ws.org/Vol-1422/186.pdf.

Z. Pitra, L. Bajer, and M. Holeňa. Doubly trained evolution control for the surrogate cma-es. In J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, pages 59–68, Cham, 2016. Springer International Publishing. URL https://link.springer.com/chapter/10.1007/978-3-319-45823-6_6.

Z. Pitra, L. Bajer, J. Repický, and M. Holeňa. Adaptive Doubly Trained Evolution Control for the Covariance Matrix Adaptation Evolution Strategy. In *ITAT 2017: Information Technologies–Applications and Theory*, ITAT 2017, North Charleston, USA, 2017a. CreateSpace Independent Publishing Platform.

Z. Pitra, L. Bajer, J. Repický, and M. Holeňa. Comparison of ordinal and metric Gaussian process regression as surrogate models for CMA evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, pages 1764–1771, New York, NY, USA, 2017b. ACM. doi: 10.1145/3067695.3084206.

Z. Pitra, L. Bajer, J. Repický, and M. Holeňa. Overview of surrogate-model versions of covariance matrix adaptation evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, pages 1622–1629, New York, NY, USA, 2017c. ACM. doi: 10.1145/3067695.3082539.

Z. Pitra, J. Repický, and M. Holeňa. Boosted regression forest for the doubly trained surrogate covariance matrix adaptation evolution strategy. In S. Krajci, editor, *Proceedings of the 18th Conference Information Technologies - Applications and Theory (ITAT 2018), Hotel Plejsy, Slovakia, September 21-25, 2018*, volume 2203 of *CEUR Workshop Proceedings*, pages 72–79, North Charleston, USA, 2018a. CEUR-WS.org.

Z. Pitra, J. Repický, and M. Holeňa. Transfer of knowledge for surrogate model selection in cost-aware optimization. In G. Krempl, V. Lemaire, D. Kottke, A. Calma, A. Holzinger, R. Polikar, and B. Sick, editors, *Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning (ECML 2018) and Principles and Practice of Knowledge Discovery in Databases (PKDD 2018), Dublin, Ireland, September 10th, 2018*, volume 2192 of *CEUR Workshop Proceedings*, pages 89–94. CEUR-WS.org, 2018b. URL http://ceur-ws.org/Vol-2192/ialatecml_paper9.pdf.

Z. Pitra, L. Bajer, and M. Holeňa. Knowledge-based selection of Gaussian process surrogates. In D. Kottke, V. Lemaire, A. Calma, G. Krempl, and A. Holzinger, editors, *ECML PKDD 2019: Workshop & Tutorial on Interactive Adaptive Learning. Proceedings*, ECML PKDD 2019, pages 48–63, Würzburg, Germany, 2019a.

Z. Pitra, J. Repický, and M. Holeňa. Landscape analysis of Gaussian process surrogates for the covariance matrix adaptation evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pages 691–699, New York, NY, USA, 2019b. Association for Computing Machinery.

Z. Pitra, M. Hanuš, J. Koza, J. Tumpach, and M. Holeňa. Interaction between model and its evolution control in surrogate-assisted CMA evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, pages 528–536, New York, NY, USA, 2021. Association for Computing Machinery.

Z. Pitra, J. Koza, J. Tumpach, and M. Holeňa. Landscape analysis for surrogate models in the evolutionary black-box context. pages 1–34, 2022. doi: 10.48550/ARXIV.2203.11315. URL https://arxiv.org/abs/2203.11315. Under review in journal. Preprint available on arXiv.org.

E. Pitzer and M. Affenzeller. *A Comprehensive Survey on Fitness Landscape Analysis*, pages 161–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

M. J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large-Scale Nonlinear Optimization*, number 83 in Nonconvex Optimization and Its Applications, pages 255–297. Springer US, 2006.

M. J. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report NA2009/06, University of Cambridge, Cambridge, UK, 2009.

C. E. Rasmussen and H. Nickisch. GPML 4.0. matlab toolbox. http://www.gaussianprocess.org/gpml/code/matlab/doc/.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. MIT Press, 2006.

A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In A. Eiben, T. Bäck, M. Schoenauer, and S. H.-P., editors, *Parallel Problem Solving from Nature*, pages 87–96. Springer, 1998.

I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.

Q. Renau, J. Dréo, C. Doerr, and B. Doerr. Expressiveness and robustness of landscape features. In M. López-Ibáñez, A. Auger, and T. Stützle, editors, *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, pages 2048–2051. ACM, 2019.

Q. Renau, C. Doerr, J. Dréo, and B. Doerr. Exploratory landscape analysis is strongly sensitive to the sampling strategy. In T. Bäck, M. Preuss, A. H. Deutz, H. Wang, C. Doerr, M. T. M. Emmerich, and H. Trautmann, editors, *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II*, volume 12270 of *Lecture Notes in Computer Science*, pages 139–153. Springer, 2020.

Q. Renau, J. Dréo, C. Doerr, and B. Doerr. Towards explainable exploratory landscape analysis: Extreme feature selection for classifying BBOB functions. In P. A. Castillo and J. L. J. Laredo, editors, *Applications of Evolutionary Computation - 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings*, volume 12694 of *Lecture Notes in Computer Science*, pages 17–33. Springer, 2021.

J. Repický, L. Bajer, Z. Pitra, and M. Holeňa. Adaptive generation-based evolution control for Gaussian process surrogate models. In J. Hlaváčová, editor, *Proceedings of the 17th Conference on Information Technologies - Applications and Theory (ITAT 2017), Martinské hole, Slovakia, September 22-26, 2017*, volume 1885 of *CEUR Workshop Proceedings*, pages 136–143. CEUR-WS.org, 2017. URL http://ceur-ws.org/Vol-1885/136.pdf.

J. Repický, M. Holeňa, and Z. Pitra. Automated selection of covariance function for Gaussian process surrogate models. In S. Krajci, editor, *Proceedings of the 18th Conference Information Technologies - Applications and Theory (ITAT 2018), Hotel Plejsy, Slovakia, September 21-25, 2018*, volume 2203 of *CEUR Workshop Proceedings*, pages 64–71. CEUR-WS.org, 2018a. URL http://ceur-ws.org/Vol-2203/64.pdf.

J. Repický, Z. Pitra, and M. Holeňa. Adaptive selection of Gaussian process model for active learning in expensive optimization. In G. Krempl, V. Lemaire, D. Kottke, A. Calma, A. Holzinger, R. Polikar, and B. Sick, editors, *Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning (ECML 2018) and Principles and Practice of Knowledge Discovery in Databases (PKDD 2018), Dublin, Ireland, September 10th, 2018*, volume 2192 of *CEUR Workshop Proceedings*, pages 80–84. CEUR-WS.org, 2018b. URL http://ceur-ws.org/Vol-2192/ialatecml_paper7.pdf.

J. R. Rice. The algorithm selection problem*. volume 15 of *Advances in Computers*, pages 65–118. Elsevier, 1976. doi: http://dx.doi.org/10.1016/S0065-2458(08)60520-3.

L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2012.

T. P. Runarsson. *Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models*, pages 401–410. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. doi: 10.1007/978-3-540-30217-9_41.

J. Růžička, J. Koza, J. Tumpach, Z. Pitra, and M. Holeňa. Combining Gaussian processes with neural networks for active learning in optimization. In *ECML PKDD 2021: Workshop on Interactive Adaptive Learning*, pages 105–120, 2021. URL https://www.activeml.net/ial2021/pdf/ialatecml_paper9.pdf.

B. S. Saini, M. Lopez-Ibanez, and K. Miettinen. Automatic surrogate modelling technique selection based on features of optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pages 1765–1772, New York, NY, USA, 2019. Association for Computing Machinery.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

H.-P. Schwefel. Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Diploma thesis, Technical University of Berlin, 1965.

H.-P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

B. Schweizer and E. F. Wolff. On nonparametric measures of dependence for random variables. *Annals of Statistics*, 9(4):879–885, 1981.

D. Spiegelhalter, N. Best, B. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 64(4):583–616, 2002.

P. K. Srijith, S. Shevade, and S. Sundararajan. *A Probabilistic Least Squares Approach to Ordinal Regression*, pages 683–694. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012a. doi: 10.1007/978-3-642-35101-3_58.

P. Srijith, S. Shevade, and S. Sundararajan. Validation based sparse gaussian processes for ordinal regression. In *ICONIP 2012*, pages 409–416, 2012b.

C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2017. doi: 10.1109/TEVC.2017.2675628.

S. Sundararajan and S. S. Keerthi. Predictive approaches for choosing hyperparameters in Gaussian processes. *Neural Computation*, 13(5):1103–1118, 2001. doi: 10.1162/08997660151134343.

S. Swarzberg, G. Seront, and H. Bersini. S.T.E.P.: The easiest way to optimize a function. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, volume 1, pages 519–524, 1994. doi: 10.1109/ICEC.1994.349896.

H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by Gaussian processes with improved preselection criterion. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, volume 1, pages 692–699 Vol.1, 2003.

H. Ulmer, F. Streichelt, and A. Zell. Evolution strategies with controlled model assistance. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 2, pages 1569–1576, 2004. doi: 10.1109/CEC.2004.1331083.

University of California, Irvine. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn, 2016.

V. Volz, G. Rudolph, and B. Naujoks. Investigating Uncertainty Propagation in Surrogate-assisted Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 881–888, New York, 2017. ACM.

S. Watanabe. Almost all learning machines are singular. pages 383–388, 2007.

S. Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *J. Mach. Learn. Res.*, 11:3571–3594, 2010.

A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1067–1075, Atlanta, Georgia, USA, 2013. PMLR.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 2016. PMLR.

J. Wu, S. Shekh, N. Y. Sergiienko, B. S. Cazzolato, B. Ding, F. Neumann, and M. Wagner. Fast and effective optimisation of arrays of submerged wave energy converters. In T. Friedrich, F. Neumann, and A. M. Sutton, editors, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, Denver, CO, USA, July 20 - 24, 2016*, pages 1045–1052. ACM, 2016. doi: 10.1145/2908812.2908844. URL https://doi.org/10.1145/2908812.2908844.

H. Yu, Y. Tan, C. Sun, J. Zeng, and Y. Jin. An adaptive model selection strategy for surrogate-assisted particle swarm optimization algorithm. SSCI '16, pages 1–8, 2016.

M. Zaefferer, D. Gaida, and T. Bartz-Beielstein. Multi-fidelity modeling and optimization of biogas plants. *Applied Soft Computing*, 48:13–28, 2016.

Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(1):66–76, 2007. doi: 10.1109/TSMCC.2005.855506.

A

LANDSCAPE FEATURES DEFINITIONS

Let us consider an input set of N points $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathcal{X}\}_{i=1}^{N}$, where $\mathcal{X}$ is the input space defined as follows

$$\mathcal{X} = \bigtimes_{j=1}^{D} [l_j, u_j], l_j, u_j \in \mathbb{R}, l_j < u_j \,.$$

Then we can define a sample set $\mathcal{S}$ of N pairs of observations

$$\mathcal{S} = \{(\mathbf{x}_i, y_i) \,|\, \mathbf{x}_i \in \mathbf{X}, y_i \in \mathbb{R} \cup \{\circ\}, i = 1, \ldots, N\} \,,$$

where $\circ$ denotes missing y_i value (e. g., $\mathbf{x}_i$ was not evaluated yet). The y_i values can be gathered to a set $\mathbf{y} = \{y_i\}_{i=1}^{N}$. The input space $\mathcal{X}$ can be further discretized into K blocks (cells) per dimension, such that

$$\boldsymbol{\mathcal{C}} = \left\{ \mathcal{C}_{m_1,\ldots,m_D} \,\middle|\, \mathcal{C}_{m_1,\ldots,m_D} = \bigtimes_{j=1}^{D} \left[l_j + (m_j - 1)\frac{u_j - l_j}{K}, l_j + m_j \frac{u_j - l_j}{K} \right], m_i \in \{1, \ldots, K\}, i = 1, \ldots, D \right\} .$$

The following features were reimplemented in Matlab according to the R-package `flacco` by Kerschke and Dagefoerde (2017). In the words of Kerschke (2017a), *"It is important to notice that, independent of the research domain, most of these features do not provide intuitively understandable numbers. Therefore, we strongly recommend not to interpret them on their own. Moreover, some of them are stochastic and hence should be evaluated multiple times on an instance and afterwards be aggregated in a reasonable manner. Nevertheless, they definitely provide information that can be of great importance to scientific models, such as machine learning algorithms in general or algorithm selectors in particular."* Following the remark about stochasticity, we have fixed random seeds which are not fixed in the original implementation to always return the same values for identical input.

A.1 BASIC FEATURES Φ_{Basic}

Kerschke and Dagefoerde (2017) summarized the following features providing obvious information about the input space:

- **Dimension** of the input space $\varphi_{\texttt{dim}} = D$.
- **Number of observations** $\varphi_{\texttt{obs}} = N$.
- **Minimum** and **maximum of lower bounds** $\varphi_{\texttt{lower\_min}} = \min_{i \in D} l_i$, $\varphi_{\texttt{lower\_max}} = \max_{i \in D} l_i$.
- **Minimum** and **maximum of upper bounds** $\varphi_{\texttt{upper\_min}} = \min_{i \in D} u_i$, $\varphi_{\texttt{upper\_max}} = \max_{i \in D} u_i$.
- **Minimum** and **maximum of y values** $\varphi_{\texttt{objective\_min}} = \min_{i \in N} y_i$, $\varphi_{\texttt{objective\_max}} = \max_{i \in N} y_i$.
- **Minimum** and **maximum of cell blocks per dimension** $\varphi_{\texttt{blocks\_min}}$, $\varphi_{\texttt{blocks\_max}}$.
- **Total number of cells** $\varphi_{\texttt{cells\_total}}$.
- **Number of filled cells** $\varphi_{\texttt{cells\_filled}}$.
- Binary flag stating whether the **objective function** should be **minimized** $\varphi_{\texttt{minimize\_fun}}$.

A.2 CM ANGLE FEATURES $\Phi_{\text{CM-Angl}}$

CM Angle features by Kerschke et al. (2014) extract information based on the location of the best and worst point of the cell $\mathcal{C} \in \boldsymbol{\mathcal{C}}$ considering the cell center.

Let us denote the cell center $\mathbf{z}_{\mathcal{C}}^{\text{center}} \in \mathcal{C}$, the best point within the cell $\mathbf{x}_{\mathcal{C}}^{\text{best}} \in \mathbf{X}$, where $(\mathbf{x}_{\mathcal{C}}^{\text{best}}, y_{\mathcal{C}}^{\text{best}}) \in \mathcal{S}$, $y_{\mathcal{C}}^{\text{best}} = \min\{y | (\mathbf{x}, y) \in \mathcal{S} \;\&\; \mathbf{x} \in \mathcal{C}\}$, and the worst point within the cell $\mathbf{x}_{\mathcal{C}}^{\text{worst}} \in \mathbf{X}$, where $(\mathbf{x}_{\mathcal{C}}^{\text{worst}}, y_{\mathcal{C}}^{\text{worst}}) \in \mathcal{S}$, $y_{\mathcal{C}}^{\text{worst}} = \max\{y | (\mathbf{x}, y) \in \mathcal{S} \;\&\; \mathbf{x} \in \mathcal{C}\}$.

The set of all distances from a cell center to the best point within the cell:

$$D_{\text{ctr2best}} = \{\text{dist}(\mathbf{x}_{\mathcal{C}}^{\text{best}}, \mathbf{z}_{\mathcal{C}}^{\text{center}}) \,|\, \mathbf{x}_{\mathcal{C}}^{\text{best}}, \mathbf{z}_{\mathcal{C}}^{\text{center}} \in \mathcal{C}, \forall \mathcal{C} \in \boldsymbol{\mathcal{C}}\}\,. \tag{107}$$

The set of all distances from a cell center to the worst point within the cell:

$$D_{\text{ctr2worst}} = \{\text{dist}(\mathbf{x}_{\mathcal{C}}^{\text{worst}}, \mathbf{z}_{\mathcal{C}}^{\text{center}}) \,|\, \mathbf{x}_{\mathcal{C}}^{\text{worst}}, \mathbf{z}_{\mathcal{C}}^{\text{center}} \in \mathcal{C}, \forall \mathcal{C} \in \boldsymbol{\mathcal{C}}\}\,. \tag{108}$$

The set of all angles between the best, the worst, and the cell-center point within the cell:

$$D_{\text{angle}} = \left\{ \left| \arccos \frac{(\mathbf{x}_{\mathcal{C}}^{\text{best}} - \mathbf{z}_{\mathcal{C}}^{\text{center}}) \cdot (\mathbf{x}_{\mathcal{C}}^{\text{worst}} - \mathbf{z}_{\mathcal{C}}^{\text{center}})}{\|\mathbf{x}_{\mathcal{C}}^{\text{best}} - \mathbf{z}_{\mathcal{C}}^{\text{center}}\| \|\mathbf{x}_{\mathcal{C}}^{\text{worst}} - \mathbf{z}_{\mathcal{C}}^{\text{center}}\|} \right| \,\middle|\, \mathcal{C} \in \boldsymbol{\mathcal{C}} \right\}. \tag{109}$$

The set of all differences between the best and the worst objective values per the cell normalized by the difference between overall best and worst objectives

$$D_{\text{best2worst}} = \left\{ \frac{y_{\mathcal{C}}^{\text{worst}} - y_{\mathcal{C}}^{\text{best}}}{\max \mathbf{y} - \min \mathbf{y}} \,\middle|\, \forall \mathcal{C} \in \boldsymbol{\mathcal{C}} \right\}. \tag{110}$$

- **Mean** and **standart deviation of distances from cell center to the best point within the cell** $\varphi^{\text{CM-Angl}}_{\texttt{dist\_ctr2best\_mean}}(\mathcal{S}) = \text{mean}\, D_{\text{ctr2best}}$ and $\varphi^{\text{CM-Angl}}_{\texttt{dist\_ctr2best\_std}}(\mathcal{S}) = \text{std}\, D_{\text{ctr2best}}$.
- **Mean** and **standart deviation of distances from cell center to the worst point within the cell** $\varphi^{\text{CM-Angl}}_{\texttt{dist\_ctr2worst\_mean}}(\mathcal{S}) = \text{mean}\, D_{\text{ctr2worst}}$ and $\varphi^{\text{CM-Angl}}_{\texttt{dist\_ctr2worst\_std}}(\mathcal{S}) = \text{std}\, D_{\text{ctr2worst}}$.
- **Mean** and **standart deviation of angles between the best, the worst, and the cell-center point within the cell** $\varphi^{\text{CM-Angl}}_{\texttt{angle\_mean}}(\mathcal{S}) = \text{mean}\, D_{\text{angle}}$ and $\varphi^{\text{CM-Angl}}_{\texttt{angle\_std}}(\mathcal{S}) = \text{std}\, D_{\text{angle}}$.
- **Mean** and **standart deviation of differences between the best and the worst objective value per cell** normalized by the difference between overall best and worst objectives $\varphi^{\text{CM-Angl}}_{\texttt{y\_best2worst\_mean}}(\mathcal{S}) = \text{mean}\, D_{\text{best2worst}}$ and $\varphi^{\text{CM-Angl}}_{\texttt{y\_best2worst\_std}}(\mathcal{S}) = \text{std}\, D_{\text{best2worst}}$.

A.3 CM CONVEXITY FEATURES $\Phi_{\text{CM-Conv}}$

CM Convexity features by Kerschke et al. (2014) aggregate the (estimated) convexity based on representative observations of successive cells. Each cell is represented by the sample observation $(\mathbf{x}_{\mathcal{C}}, y_{\mathcal{C}})$, $\mathbf{x}_{\mathcal{C}} \in \mathcal{C}$, that is located closest to the corresponding cell center. Let us define a block of three successive cells in one dimension $[\mathcal{C}_{\text{L}}, \mathcal{C}_{\text{C}}, \mathcal{C}_{\text{R}}] \in \mathcal{B}$, where $\mathcal{B}$ is a set of all possible blocks on $\boldsymbol{\mathcal{C}}$.

- **Estimated probability of soft concavity** $\varphi^{\text{CM-Conv}}_{\texttt{concave\_soft}}(\mathcal{S}) = \dfrac{\left|\left\{[\mathcal{C}_{\text{L}}, \mathcal{C}_{\text{C}}, \mathcal{C}_{\text{R}}] \,|\, y_{\mathcal{C}_{\text{C}}} > \frac{y_{\mathcal{C}_{\text{L}}} + y_{\mathcal{C}_{\text{R}}}}{2}\right\}\right|}{|\mathcal{B}|}$.
- **Estimated probability of hard concavity** $\varphi^{\text{CM-Conv}}_{\texttt{concave\_hard}}(\mathcal{S}) = \dfrac{\left|\left\{[\mathcal{C}_{\text{L}}, \mathcal{C}_{\text{C}}, \mathcal{C}_{\text{R}}] \,|\, y_{\mathcal{C}_{\text{C}}} > \max\{y_{\mathcal{C}_{\text{L}}}, y_{\mathcal{C}_{\text{R}}}\}\right\}\right|}{|\mathcal{B}|}$.
- **Estimated probability of soft convexity** $\varphi^{\text{CM-Conv}}_{\texttt{convex\_soft}}(\mathcal{S}) = \dfrac{\left|\left\{[\mathcal{C}_{\text{L}}, \mathcal{C}_{\text{C}}, \mathcal{C}_{\text{R}}] \,|\, y_{\mathcal{C}_{\text{C}}} < \frac{y_{\mathcal{C}_{\text{L}}} + y_{\mathcal{C}_{\text{R}}}}{2}\right\}\right|}{|\mathcal{B}|}$.
- **Estimated probability of hard convexity** $\varphi^{\text{CM-Conv}}_{\texttt{convex\_hard}}(\mathcal{S}) = \dfrac{\left|\left\{[\mathcal{C}_{\text{L}}, \mathcal{C}_{\text{C}}, \mathcal{C}_{\text{R}}] \,|\, y_{\mathcal{C}_{\text{C}}} < \min\{y_{\mathcal{C}_{\text{L}}}, y_{\mathcal{C}_{\text{R}}}\}\right\}\right|}{|\mathcal{B}|}$.

A.4 CM GRADIENT HOMOGENEITY FEATURES $\Phi_{\text{CM-Grad}}$

CM Gradient homogeneity features by Kerschke et al. (2014) aggregate the cell-wise information on the gradients between each point of a cell and its corresponding nearest neighbor. For each point in cell $\mathbf{x}_i \in \mathcal{C}$, $i = 1, \ldots, N_{\mathcal{C}}$, the gradient towards its nearest neighbor is normalized and pointed towards the one of neighbors with better y-value. The normalized sum of gradients in cell $\mathcal{C}$ is than

$$GH_{\mathcal{C}} = \frac{\left\| \sum_{i,j=1}^{N_{\mathcal{C}}} (2\mathbb{I}(y_i > y_j) - 1) \frac{\mathbf{x}_i - \mathbf{x}_j}{\text{dist}(\mathbf{x}_i, \mathbf{x}_j)} \right\|}{N_{\mathcal{C}}}. \tag{111}$$

- **Mean** and **standard deviation of homogeneity gradients** accross all cells $\varphi^{\text{CM-Grad}}_{\texttt{grad\_mean}}(\mathcal{S}) = \text{mean}_{\mathcal{C} \in \boldsymbol{\mathcal{C}}} GH_{\mathcal{C}}$ and $\varphi^{\text{CM-Grad}}_{\texttt{grad\_std}}(\mathcal{S}) = \text{std}_{\mathcal{C} \in \boldsymbol{\mathcal{C}}} GH_{\mathcal{C}}$.

A.5 CMA FEATURES Φ_{CMA}

In each CMA-ES generation g during the fitness evaluation step (Algorithm 1, Step 4), we have defined the following features in Pitra et al. (2019b) for the set of points $\mathbf{X}$:

- **Generation number** $\varphi^{\text{CMA}}_{\texttt{generation}} = g$ indicates the phase of the optimization process.
- **Step-size** $\varphi^{\text{CMA}}_{\texttt{step\_size}} = \sigma^{(g)}$ provides an information about the extent of the approximated region.
- **Number of restarts** $\varphi^{\text{CMA}}_{\texttt{restart}} = n_{\text{r}}^{(g)}$ performed till generation g may indicate landscape difficulty.
- **Mahalanobis mean distance** of the CMA-ES mean $\mathbf{m}^{(g)}$ to the sample mean $\mu_{\mathbf{X}}$ of $\mathbf{X}$

$$\varphi^{\text{CMA}}_{\texttt{mean\_dist}}(\mathbf{X}) = \sqrt{(\mathbf{m}^{(g)} - \mu_{\mathbf{X}})^{\top} \mathbf{C}_{\mathbf{X}}^{-1} (\mathbf{m}^{(g)} - \mu_{\mathbf{X}})}, \tag{112}$$

  where $\mathbf{C}_{\mathbf{X}}$ is the sample covariance of $\mathbf{X}$. This feature indicates suitability of $\mathbf{X}$ for model training from the point of view of the current state of the CMA-ES algorithm.
- Square of the $\mathbf{p}_c$ **evolution path** length $\varphi^{\text{CMA}}_{\texttt{evopath\_c\_norm}} = \left\| \mathbf{p}_c^{(g)} \right\|^2$ is the only possible non-zero eigenvalue of *rank-one update* covariance matrix $\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)\top}$ (see Section 2.2.2). That feature providing information about the correlations between consecutive CMA-ES steps indicates a similarity of function landscapes among subsequent generations.
- $\mathbf{p}_\sigma$ **evolution path** ratio, i. e., the ratio between the evolution path length $\left\| \mathbf{p}_\sigma^{(g)} \right\|$ and the expected length of a random evolution path used to update step-size. It provides a useful information about distribution changes:

$$\varphi^{\text{CMA}}_{\texttt{evopath\_s\_norm}} = \frac{\left\| \mathbf{p}_\sigma^{(g)} \right\|}{E \left\| \mathcal{N}(\mathbf{0}, \mathbf{I}) \right\|} = \frac{\left\| \mathbf{p}_\sigma^{(g)} \right\| \Gamma\left(\frac{D}{2}\right)}{\sqrt{2}\, \Gamma\left(\frac{D+1}{2}\right)}. \tag{113}$$

- **CMA similarity likelihood**. The log-likelihood of the set of points $\mathbf{X}$ with respect to the CMA-ES distribution may also serve as a measure of its suitability for training

$$\begin{aligned} \varphi^{\text{CMA}}_{\texttt{cma\_lik}}(\mathbf{X}) = &- \frac{N}{2} \left(D \ln 2\pi {\sigma^{(g)}}^2 + \ln \det \mathbf{C}^{(g)} \right) \\ &- \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{X}} \left(\frac{\mathbf{x} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \right)^{\top} {\mathbf{C}^{(g)}}^{-1} \left(\frac{\mathbf{x} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \right). \end{aligned} \tag{114}$$

A.6 DISPERSION FEATURES Φ_{Dis}

The dispersion features by Lunacek and Whitley (2006) compare the dispersion among observations from $\mathcal{S}$ and among a subset of these points. The subsets are created based on tresholds using the `quantiles` 0.02, 0.05, 0.1, and 0.25 of the objective values $\mathbf{y}$.

The set of all distances within the set $\mathcal{S}$:

$$D_{\text{all}} = \{\text{dist}(\mathbf{x}_i, \mathbf{x}_j) \,|\, i, j = 1, \ldots, N\}\,. \tag{115}$$

The quantile subset of all distances:

$$D_{\texttt{quantile}} = \{\text{dist}(\mathbf{x}_i, \mathbf{x}_j) \,|\, y_i, y_j <= Q_{\texttt{quantile}}(\mathbf{y}),\}\,. \tag{116}$$

◇ **Ratio of** the quantile subset and all points **median distances**

$$\varphi^{\text{Dis}}_{\texttt{ratio\_median\_[quantile]}}(\mathcal{S}) = \frac{\text{median}\, D_{\texttt{quantile}}}{\text{median}\, D_{\text{all}}}\,. \tag{117}$$

◇ **Ratio of** the quantile subset and all points **mean distances**

$$\varphi^{\text{Dis}}_{\texttt{ratio\_mean\_[quantile]}}(\mathcal{S}) = \frac{\text{mean}\, D_{\texttt{quantile}}}{\text{mean}\, D_{\text{all}}}\,. \tag{118}$$

◇ **Difference between** the quantile subset and all points **median distances**

$$\varphi^{\text{Dis}}_{\texttt{diff\_median\_[quantile]}}(\mathcal{S}) = \text{median}\, D_{\texttt{quantile}} - \text{median}\, D_{\text{all}}\,. \tag{119}$$

◇ **Difference between** the quantile subset and all points **mean distances**

$$\varphi^{\text{Dis}}_{\texttt{diff\_mean\_[quantile]}}(\mathcal{S}) = \text{mean}\, D_{\texttt{quantile}} - \text{mean}\, D_{\text{all}}\,. \tag{120}$$

A.7 INFORMATION CONTENT FEATURES Φ_{Inf}

The Information Content of Fitness Sequences by Muñoz et al. (2015) approach is based on a symbol sequence $\Psi = \{\psi_1, \ldots, \psi_{N-1}\}$, where

$$\psi_i = \begin{cases} \bar{1}\,, & \text{if} \quad \frac{y_{i+1}-y_i}{\|\mathbf{x}_{i+1}-\mathbf{x}_i\|} < -\varepsilon\,, \\ 0\,, & \text{if} \quad \left|\frac{y_{i+1}-y_i}{\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}\right| \leq \varepsilon\,, \\ 1\,, & \text{if} \quad \frac{y_{i+1}-y_i}{\|\mathbf{x}_{i+1}-\mathbf{x}_i\|} > \varepsilon\,. \end{cases} \tag{121}$$

The sequence considers observations from a sample set $\mathcal{S}$ as a random walk accross the objective landscape and depends on the information sensitivity parameter $\varepsilon > 0$.

The symbol sequence Ψ is aggregated by the information content $H(\varepsilon) = -\sum_{i \neq j} p_{ij} \log_6 p_{ij}$, where p_{ij} is the probability of having the "block" $\psi_i\psi_j$, where $\psi_i, \psi_j \in \{\bar{1}, 0, 1\}$, within the sequence. Note that the base of the logarithm was set to six as this equals the number of possible blocks $\psi_i\psi_j$ for which $\psi_i \neq \psi_j$, i. e., $\psi_i\psi_j \in \{\bar{1}0, 0\bar{1}, \bar{1}1, 1\bar{1}, 01, 10\}$.

Another aggregation of the information is the so-called partial information content $M(\varepsilon) = |\Psi'|/(N-1)$, where Ψ' is the symbol sequence of alternating $\bar{1}$'s and 1's, which is derived from Ψ by removing all 0 and repeated symbols.

When we do consider $\circ$ as a valid state of y, the sequence Ψ consists of $\psi_i \in \{\bar{1}, 0, 1, \bar{N}\}$, where $\psi_i = \bar{N}$, if $y_{i+1} = \circ$ or $y_i = \circ$. Thus, $H(\varepsilon) = -\sum_{i \neq j} p_{ij} \log_{12} p_{ij}$ due to the increased number of possible $\psi_i\psi_j$ blocks, i. e., $\psi_i\psi_j \in \{\bar{1}0, 0\bar{1}, \bar{1}1, 1\bar{1}, 01, 10, \bar{N}\bar{1}, \bar{1}\bar{N}, \bar{N}0, 0\bar{N}, \bar{N}1, 1\bar{N}\}$. Therefore, features based on Ψ' can be utilized only when $\circ$ state is not present in $\mathbf{y}$.

Based on sequences Ψ and Ψ' the following features can be defined according to Muñoz et al. (2015):

- ◇ **Maximum information content** $\varphi^{\text{Inf}}_{\texttt{h\_max}}(\mathcal{S}) = \max_\varepsilon H(\varepsilon)$.
- ◇ **Settling sensitivity** $\varphi^{\text{Inf}}_{\texttt{eps\_s}}(\mathcal{S}) = \log_{10} \min(\varepsilon | H(\varepsilon) < s)$, where default $s = 0.05$ (see (Muñoz et al., 2015)).
- ◇ **Maximum sensitivity** $\varphi^{\text{Inf}}_{\texttt{eps\_max}}(\mathcal{S}) = \arg\max_\varepsilon H(\varepsilon)$.
- ◇ **Initial partial information** $\varphi^{\text{Inf}}_{\texttt{m0}}(\mathcal{S}) = M(\varepsilon = 0)$.
- ◇ **Ratio of partial information sensitivity** $\varphi^{\text{Inf}}_{\texttt{eps\_ratio}}(\mathcal{S}) = \log_{10} \max(\varepsilon | M(\varepsilon) > r\varphi^{\text{Inf}}_{\texttt{m0}}(\mathcal{S})))$, where default $r = 0.5$ (see also (Muñoz et al., 2015)).

A.8 LEVELSET FEATURES Φ_{Lvl}

In Levelset features by Mersmann et al. (2011), the sample set $\mathcal{S}$ is split into two classes by a specific treshold calculated using the quantiles 0.1, 0.25, and 0.5. Linear, quadratic, and mixture discriminant analysis (`lda`, `qda`, and `mda`) are used to predict whether the objective values $\mathbf{y}$ fall below or exceed the calculated threshold. The extracted features are based on the distribution of the resulting cross-validated mean misclassification errors of each classifier.

- ◇ **Mean misclassification error** of appropriate discriminant analysis `method` using defined `quantile` $\varphi^{\text{Lvl}}_{\texttt{mmce\_[method]\_[quantile]}}(\mathcal{S})$.
- ◇ **Ratio between** mean misclassification **errors** of two discriminant analysis `methods` using a given `quantile` $\varphi^{\text{Lvl}}_{\texttt{[method1]\_[method2]\_[quantile]}}(\mathcal{S})$.

A.9 METAMODEL FEATURES Φ_{MM}

To calculate metamodel features by Mersmann et al. (2011), linear and quadratic regression models (`lin` and `quad`) with or without interactions are fitted to a sample set $\mathcal{S}$. The adjusted coefficient of determination R^2 and features reflecting the size relations of the model coefficients are extracted:

- ◇ **Adjusted** R^2 of a **simple model** $\varphi^{\text{MM}}_{\texttt{[model]\_simple\_adj\_r2}}(\mathcal{S})$.
- ◇ **Adjusted** R^2 of a **model with interactions** $\varphi^{\text{MM}}_{\texttt{[model]\_w\_interact\_adj\_r2}}(\mathcal{S})$.
- ◇ **Intercept** of a simple **linear model** $\varphi^{\text{MM}}_{\texttt{lin\_simple\_intercept}}(\mathcal{S})$.
- ◇ **Minimal** absolute **value of linear model coefficients** $\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_min}}(\mathcal{S})$.
- ◇ **Maximal** absolute **value of linear model coefficients** $\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max}}(\mathcal{S})$.
- ◇ **Ratio** of **maximal** and **minimal** absolute **value of linear model coefficients** $\varphi^{\text{MM}}_{\texttt{lin\_simple\_coef\_max\_by\_min}}(\mathcal{S})$.
- ◇ **Ratio** of **maximal** and **minimal** absolute **value of quadratic model coefficients** $\varphi^{\text{MM}}_{\texttt{quad\_simple\_cond}}(\mathcal{S})$.

A.10 NEAREST BETTER CLUSTERING FEATURES Φ_{NBC}

Nearest better clustering features by Kerschke et al. (2015) extract information based on the comparison of the sets of distances from all observations towards their nearest neighbors (D_{nn}) and their nearest better neighbors (D_{nb}).

The distance to the nearest neighbor of a search point $\mathbf{x}_i$, $i = 1, \ldots, N$ from a set of points $\mathbf{X}$:

$$d_{\text{nn}}(\mathbf{x}_i, \mathbf{X}) = \min(\text{dist}(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_j \in \mathbf{X}, i \neq j)\,. \tag{122}$$

The distance to the nearest better neighbor:

$$d_{\text{nb}}(\mathbf{x}_i, \mathbf{X}) = \min(\text{dist}(\mathbf{x}_i, \mathbf{x}_j) | y_j < y_i, \mathbf{x}_j \in \mathbf{X}))\,. \tag{123}$$

The set of all nearest neighbor distances within the set $\mathbf{X}$:

$$D_{\text{nn}} = \{d_{\text{nn}}(\mathbf{x}_i, \mathbf{X}) | i = 1, \ldots, N\}\,. \tag{124}$$

The set of all nearest better distances:

$$D_{\text{nb}} = \{d_{\text{nb}}(\mathbf{x}_i, \mathbf{X}) | i = 1, \ldots, N\}\,. \tag{125}$$

The features are defined as follows:

- ◇ **Ratio of the standard deviations** between the D_{nn} and D_{nb} $\varphi^{\text{NBC}}_{\texttt{nb\_std\_ratio}}(\mathcal{S}) = \frac{\text{std}\, D_{\text{nn}}}{\text{std}\, D_{\text{nb}}}$.
- ◇ **Ratio of the means** between the D_{nn} and D_{nb} $\varphi^{\text{NBC}}_{\texttt{nb\_mean\_ratio}}(\mathcal{S}) = \frac{\text{mean}\, D_{\text{nn}}}{\text{mean}\, D_{\text{nb}}}$.
- ◇ **Correlation between the distances** of the nearest neighbors and nearest better neighbors $\varphi^{\text{NBC}}_{\texttt{nb\_cor}}(\mathcal{S}) = \text{corr}(D_{\text{nn}}, D_{\text{nb}})$.
- ◇ **Coefficient of** variation of the **distance ratios** $\varphi^{\text{NBC}}_{\texttt{dist\_ratio}}(\mathcal{S}) = \frac{\text{std}\, \mathcal{W}}{\text{mean}\, \mathcal{W}}$, where $\mathcal{W} = \left\{ \frac{d_{\text{nn}}(\mathbf{x}, \mathbf{X})}{d_{\text{nb}}(\mathbf{x}, \mathbf{X})} \middle| \mathbf{x} \in \mathbf{X} \right\}$.
- ◇ **Correlation between the fitness value, and the count of observations** to whom the current observation is the nearest better neighbor $\varphi^{\text{NBC}}_{\texttt{nb\_fitness\_cor}}(\mathcal{S}) = -\text{corr}(\{\deg^-(\mathbf{x}_i), y_i | i = 1, \ldots, N\})$, where $\deg^-(\mathbf{x}_i)$ is the indegree of $\mathbf{x}_i$ in the nearest better graph (the number of points for which a certain point is the nearest better point).

A.11 PCA FEATURES Φ_{PCA}

The features from Kerschke (2017a) extract information from a principal component analysis, which is performed while including and excluding the objective values $\mathbf{y}$ and that are based on the covariance, as well as correlation matrix, respectively.

- ◇ **Proportion of principal components** that are **needed to explain** 90% of the **X's variance** $\varphi^{\text{PCA}}_{\texttt{pca\_cov\_x}}(\mathbf{X})$.
- ◇ **Proportion of principal components** that are **needed to explain** 90% of the **X's correlation** $\varphi^{\text{PCA}}_{\texttt{pca\_corr\_x}}(\mathbf{X})$.
- ◇ **Proportion of principal components** that are **needed to explain** 90% of the $[\mathbf{X}, \mathbf{y}]$**'s variance** $\varphi^{\text{PCA}}_{\texttt{pca\_cov\_init}}(\mathcal{S})$.
- ◇ **Proportion of principal components** that are **needed to explain** 90% of the $[\mathbf{X}, \mathbf{y}]$**'s correlation** $\varphi^{\text{PCA}}_{\texttt{pca\_corr\_init}}(\mathcal{S})$.
- ◇ **Percentage of variation** that is **explained** by the **first principal component** using **covariance** of $\mathbf{X}$ $\varphi^{\text{PCA}}_{\texttt{pca\_pc1\_cov\_x}}(\mathbf{X})$.
- ◇ **Percentage of variation** that is **explained** by the **first principal component** using **correlation** of $\mathbf{X}$ $\varphi^{\text{PCA}}_{\texttt{pca\_pc1\_corr\_x}}(\mathbf{X})$.
- ◇ **Percentage of variation** that is **explained** by the **first principal component** using **covariance** of $[\mathbf{X}, \mathbf{y}]$ $\varphi^{\text{PCA}}_{\texttt{pca\_pc1\_cov\_init}}(\mathcal{S})$.
- ◇ **Percentage of variation** that is **explained** by the **first principal component** using **correlation** of $[\mathbf{X}, \mathbf{y}]$ $\varphi^{\text{PCA}}_{\texttt{pca\_pc1\_corr\_init}}(\mathcal{S})$.

A.12 y-DISTRIBUTION FEATURES $\Phi_{\text{y-D}}$

The $\mathbf{y}$-distribution features from Mersmann et al. (2011) compute the basic statistics of the fitness values $\mathbf{y}$.

- ◇ **Skewness** and **kurtosis** of $\mathbf{y}$ values $\varphi^{\text{y-D}}_{\texttt{skewness}}(\mathbf{y})$ and $\varphi^{\text{y-D}}_{\texttt{kurtosis}}(\mathbf{y})$.
- ◇ **Number of peaks** of the kernel-based estimation of the density of $\mathbf{y}$-distribution $\varphi^{\text{y-D}}_{\texttt{number\_of\_peaks}}(\mathbf{y})$, where the peak is defined according to Kerschke and Dagefoerde (2017).