Suitability of Modern Neural Networks for Active and Transfer Learning in Surrogate-Assisted Black-Box Optimization

Martin Holeňa[1,2], Jan Koza[2]

[1]Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic
[2]Czech Technical University, Faculty of Information Technology, Prague, Czech Republic

Abstract

Active learning plays a crucial role in black-box optimization, especially for objective functions that are expensive to evaluate. Continuous black-box optimization has adopted an approach called surrogate modelling, where the original black-box objective is approximated with a regression model. An active learning task in this context is to decide which points should be evaluated using the original objective to update the surrogate model. Apart from low-order polynomials, the first surrogate models were artificial neural networks of the kinds multilayer perceptron and radial basis function network. In the late 2000s, neural networks have been superseded by other kinds of surrogate models, primarily Gaussian processes. However, over the last 15 years, neural networks have seen significant and successful development, suggesting that they once again have the potential to serve as promising surrogate models. This paper reviews possible research directions concerning that potential, and recalls initial results from investigations in some of these directions. Finally, it contributes to those results by investigating the state-of-the-art black-box optimizer CMA-ES surrogate-assisted by two variants of random-activation-function neural network ensembles.

1. Introduction

One area where active learning plays a really important role is *black-box optimization (BBO)*, i.e., optimization of objective functions for which no analytical description is provided. It employs optimization methods that need as input only points in the search space paired with respective values of the objective function obtained in a non-analytical way, e.g. from sensors, in experiments or through numerical simulations. Most frequently used are evolutionary optimization approaches, such as evolution strategies, genetic algorithms, and differential evolution, or other metaheuristics, such as particle swarm optimization.

Because BBO methods receive only information about values of the objective function, they typically need many such values. This is a problem in situations when evaluating the black-box objective function is time-consuming and/or expensive. That is frequently the case if it is evaluated empirically in experiments. For example, for the evolutionary optimization tasks described in the book [1], the evaluation of a comparatively small generation of a genetic algorithm can sometimes take more than a week and cost more than 10000 €. To deal with expensive evaluations, continuous BBO has in the late 1990s and early 2000s adopted an approach called *surrogate modelling* or *metamodelling* [2, 3, 4, 5, 6, 7, 8]. In principle, a surrogate model is any regression model that with a sufficient fidelity approximates the original black-box objective function, restricting the necessity of its evaluation only to a small proportion of points, whereas everywehere else, only the surrogate model is used.

Selecting the points in which the original objective function should be evaluated is a step in which active learning is involved. However, it is not active learning of a regression model although the surrogate model itself is a regression model. The reason is that its utility functions are not based on the model, like are the commoly used utility functions uncertainty decrease, model performance, diversity, or surprise-novelty. Instead, they are based on the BBO, the most common being minimizing the objective function for a given evaluation budget, and minimizing the evaluation budget for a given

IAL@ECML-PKDD'24: 8th Intl. Worksh. & Tutorial on Interactive Adaptive Learning, Sep. 9th, 2024, Vilnius, Lithuania

martin@cs.cas.cz (M. Holeňa); kozajan@fit.cvut.cz (J. Koza)

objective-function threshold. Nevertheless, even active learning in surrogate-assisted BBO follows the basic priciple of active learning: to actively select next model inputs according to the considered utility function.

The earliest kinds of surrogate models in continuous BBO were *low-order polynomials* and *artificial neural networks* (ANNs) of the kind multilayer perceptron (MLP). The former have always remained a suitable choice in situations when enough evaluations of the original black-box objective function are affordable for the approximation properties of polynomials to be in effect. On the other hand, surrogate modelling for substantially less evaluations of the original objective has during the last two decades undergone further development. MLPs were soon replaced with another kind of ANNs, radial basis function networks (RBFs), which better fit local peculiarities of an objective function landscape. Those networks, however, have since the late 2000s been superseded by other kinds of surrogate models, primarily *Gaussian processes* (GPs), but also ranking support vector machines (RSVMs), and random forests (RFs). GPs are currently the most successful kind of surrogate models for BBO with small evaluation budget of functions with complicated multimodal landscapes, mainly due to their ability to assess the uncertainty of the estimate of the original objective function in a given point, more precisely, to provide the probability distribution of this estimate. That property of GPs allows to combine the original BBO method, e.g. an evolutionary one, with Bayesian optimization.

Consequently, only little attention has been paid to ANN-based surrogate models in continuous BBO during the last 15 years. This contrasts with the intense and successful development of the ANN area during that time, which suggests that ANNs again have the potential to serve as promising surrogate models. This paper attempts to bring a small contribution to research into that potential, presenting in addition a review of possible directions for such a research, connected with different classes of neural networks. Moreover, it also points out that ANNs can serve as the basis for transfer learning between surrogate-assisted BBO of different functions.

The next section surveys important aspects and key methods concerning surrogate-assisted continuous BBO. The review of possible research directions concerning the usability of modern neural networks in surrogate-assisted BBO is presented in Section 3. Finally, Section 4 reports an experimental contribution to one of those research directions.

2. Surrogate-Assisted Continuous BBO

Surrogate modelling for continuous BBO relies on combination and interaction of three components: a *regression model* serving as a surrogate of the original black-box objective function, a *BBO method* seeking the optimum of that objective function, and a strategy when to evaluate the original objective function and when its surrogate model. That strategy is in the context of evolutionary BBO usually called *evolution control* [9, 10, 11, 12, 13]. There are two other aspects, namely observing constraints on the feasible set of the black-box objective function (cf. e.g. [14, 15]), and generalizing surrogate modelling from single objective to multiple objectives (cf. e.g. [16, 17]), however, we will restrict our attention to single-objective unconstrained optimization.

As already mentioned in the introduction, the regression models that are the most suitable kind of surrogate models if sufficiently many evaluations of the original black-box objective function are affordable, are low-order polynomials, typically quadratic functions [18, 19, 20, 21, 22]. For substantially less evaluations, the most traditional kind have been MLPs [23, 9], soon replaced with RBFs [24, 25, 26, 21, 22], and since the late 2000s with GPs a.k.a. kriging [27, 28, 11, 29, 30]. Occasionally, RBFs were used as local models in combination with GP-based global models [31]. Other kinds of surrogate models employed during the last decade include decision trees [32], RFs [33, 34, 32], and RSVMs [35, 36]. The last one has an exceptional property of *invariance with respect to order-preserving transformations* of the objective function. This is important in situations when the BBO algorithm possesses such invariance, a frequently encountered property of evolutionary algorithms. On the other hand, the surrogate modelling methods proposed in [11] and [28] use GPs to perform preselection based on a partial ordering that is also invariant with respect to order-preserving transformations. More importantly, the adaptive function value warping approach recently proposed in [37] aims at providing such invariance to any surrogate model. As a final remark to different kinds of surrogate models, important works about that topic always consider several kinds [38, 12, 39, 20, 32], to compare them and select the best among them, and in [22, 39] also to aggregate their results, thus providing a *team of surrogate models*.

As to the BBO methods, not only the two most important kinds of surrogate models, i.e. low-order polynomials [18, 19, 20], and GPs [26, 28, 11, 29, 30], but also the less common RBFs, RFs, and RSVMs [24, 36, 33, 34] are most often combined with the *Covariance matrix adaptation evolution strategy* (CMA-ES). That is not surprising because CMA-ES has already in the 2000s become a state-of-the-art approach to single-objective unconstrained continuous BBO. Basically, the CMA-ES evolves a Gaussian estimate of the position of the minimum of the original objective function. That evolution relies on simultaneous adaptation of the vector mean of the Gaussian estimate, of the scalar step size, and of the covariance matrix. For more details of this sophisticated evolution strategy, the reader is referred to the journal papers [40, 41]. GPs were also combined with other evolutionary optimization methods [27, 42], and GPs, polynomials, and RBFs were combined with particle swarm optimization [22] and with memetic optimization [25]. Moreover, GPs are used in black-box optimization in two different ways. In connection with evolutionary and similar BBO methods, they serve as a regression model evaluated instead of the original objective function. In addition, they also play a key role in *Bayesian optimization*, which then relies on GP-estimates of probability distributions of values of the original objective. Those probability distributions enable several ways of searching for optima of that objective function, each of them governed by a specific assessment of uncertainty of the objective function estimate, commonly called *acquisition function* [43, 44, 45]. Occasionally, Bayesian optimization is combined with CMA-ES. For example in [46], optimization switches from the most traditional Bayesian optimization method, EGO (Efficient Global Optimization) [43], to CMA-ES.

Finally, evolution control has been since the first surrogate-assisted BBO methods performed basically in two ways, *generation-based*, and *individual-based*. In the generation based, all points are in some generations evaluated with the true objective function, and in the remaining generations with the model. On the other hand, in every generation of the individual-based evolution control, based on the evaluation of all points with the model, a preselection of points to be evaluated with the true objective function is performed [9]. In most of the surrogate-assisted methods, however, the evolution control is specifically tailored to the respective method. Noteworthy, the authors of [13] investigated mutually replacing the evolution control of two important polynomial-assisted methods lmm-CMA [18, 19] and lq-CMA-ES [20], and of two variants of the GP-assisted method DTS-CMA-ES [47, 12] with the evolution control of the others. According to their findings, the success of those important methods is definitely not limited to using the respective specific tailored evolution control. The surrogate-assisted black-box optimization methods constructing several surrogate models simultaneously either aggregate them to a team [25, 22] or complement the evolution control by a classifier selecting the most appropriate among those models. Important examples of classifiers used in this context are ANNs [48, 49, 50], and classification trees [51, 52]. Their learning can be viewed as *metalearning* because it is based on *metafeatures*, i.e. properties empirically characterizing the objective function landscape and the BBO method [21, 32, 49, 53]. Apart from classification according to the appropriateness of the surrogate model for the considered data, metalearning can be also used for regression of model error on the combination of values of metafeatures [54].

3. Usability of Modern Neural Networks in Surrogate-Assisted BBO

This section primarily reviews eight kinds of modern neural networks that we consider worth a research into their ability to serve as surrogate models in BBO. A high-level overview of those kinds of ANNs is given in Table 1, which for each of them mentions whether such research has already started. In Subsection 3.1, two kinds integrating GPs into ANNs are recalled. Subsection 3.2 recalls three kinds of ANNs providing the most advantageous property of GPs, their ability to estimate the distribution of black-box objective function values. Finally, in Subsection 3.3, three well-known kinds of modern neural

Table 1
A high-level overview of kinds of ANNs that we consider worth a research with respect to surrogate modelling for BBO

ANNs + main references	Research into its ability to serve as surrogate model in BBO
MLPs with a GP as the final layer [55, 56]	First investigations [57, 58]
Deep GP networks [59, 60, 61, 62, 63]	Not
Tangent kernel networks [64, 65]	Not
Prior networks [66, 67, 68, 69, 70]	First investigations [71]
Ensembles of neural networks [72, 73, 74, 75, 76]	First investigations [this paper]
Variational autoencoders [77, 78]	Not
Generative adversarial networks [79, 80]	Not
Transformers [81, 82]	Not

networks, namely variational autoencoders, transformers, and generative adversarial networks, are recalled due to the fact that they have already proven useful in the related area of Bayesian optimization. In addition, Subsection 3.4 is devoted to knowledge transfer in surrogate-assisted BBO, which relates to the usability of modern neural networks through their important role in transfer learning.

3.1. Integration of GPs into ANNs

The integration of GPs into ANNs has been proposed on two different levels:

1. At the layer level – a *GP serves as the final layer of an MLP* [55, 56]. Integration on that level is based on the following two assumptions:
   (i) 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 defined. Consequently, the number n_O of neurons in the last hidden layer fulfills $\mathcal{X} \subset \mathbb{R}^{n_O}$, and the ANN maps an input v into a point $x = \mathrm{net}(v) \in \mathcal{X}$, corresponding to an observation $f(x+\varepsilon)$ governed by the GP, where ε is a zero-mean Gaussian noise. From the point of view of the ANN inputs, the GP is now $\mathcal{GP}(m_{\mathrm{GP}}(\mathrm{net}(\cdot)), \kappa(\mathrm{net}(\cdot), \mathrm{net}(\cdot)))$, where m_{GP} is the mean function, and κ is the covariance function of the GP [83].
   (ii) The GP mean μ *is assumed to be a known constant*, thus not contributing to the GP hyperparameters, and independent of net
2. At the level of individual neurons – GPs can replace all hidden and output neurons of an MLP. This kind of neural networks is commonly called *deep Gaussian process* [59, 60, 61, 62, 63, 84, 85, 86, 87, 88, 89, 90].

Integration on both levels has been developed primarily for Bayesian modelling and optimization. Nevertheless, GPs integrated as the last layer of MLPs have been used as surrogate models in a CMA-ES-driven BBO [57, 58]. In particular, those surrogate models incorporate GPs with five commonly employed covariance functions linear, quadratic, rational quadratic, squared exponential, and Matérn $\frac{5}{2}$, as well as with one composite covariance function superposing the quadratic and squared exponential. Those 6 models were compared in [57] from the point of view of regression accuracy, evaluated on a large dataset collected during many previous runs of DTS-CMA-ES on the collection of 24 noiseless benchmarks from the Comparing Continuous Optimizers platform [91, 92] (cf. Section 4) in dimensions 2, 3, 5, 10, and 20. Then in [58], they were compared on the same benchmarks in the same dimensions from the point of view of the success of surrogate-assisted optimization with CMA-ES. Unfortunately, neither of those comparisons included more traditional surrogate models nor the CMA-ES without surrogate assistance. To our knowledge, the only comparison that included both a GP integrated as the last layer of an MLP, and more traditional surrogate models, was the comparison from the point of view of regression accuracy in [93]. However, it included only one such integrated surrogate model, with the

GP using the most simple covariance function – the linear one, in addition to the traditional GP-based surrogate models with eight different covariance functions, including the five listed above.

3.2. ANNs Estimating the Distribution of Black-Box Objective Function Values

In our opinion, the property of GPs most advantageous from the point of view of surrogate modelling is that they estimate the whole distribution of a predicted value of the original black-box objective function. Recall from Section 2 that due to that property also ensembles of regression trees – RFs – are used as surrogate models [33, 34, 32]. This draws attention to those modern neural networks that also allow estimation of such a distribution. Basically, there are three classes of them, differing in the way how that estimate can be obtained.

1. The multivariate normal distribution underlying GPs is actually the *asymptotic distribution for network width increasing to infinity*. Such results have been established for several kinds of ANNs [94, 88, 95, 96, 97]. In addition, closely related is the *infinite width limit of the neural tangent kernel*, which governs the kernel gradient of the functional cost used in MLP regression [64, 65]. Although those results have great theoretical value, there can be a serious disparity between the infinite width results and their finite width counterparts [77]. Therefore, it is unclear whether they can be applied to surrogate modelling.
2. The distribution of a predicted value, or more precisely the parameters of such a distribution, can be directly learned by an ANN. The best-known kind of such neural networks are the *prior networks*, learning the parameters of a normal-inverse Wishart distribution, which is the conjugate prior to a multivariate normal distribution [66, 67, 68, 69, 70, 98]. Prior networks belong to a broader class of *evidential neural networks* [99, 100, 101, 102, 103]. Their name refers to the fact that they follow the basic principle of the Dempster-Shafer theory of evidence [104] – to fall back onto prior belief for unfamiliar data.
3. An estimate of the distribution of a predicted value is produced by an *ensemble of neural networks*. Important kinds of such ensembles are ensembles obtained through diversification of training data [105, 106], ensembles obtained through diversification of network properties [107, 108, 109], a specific subgroup of which are ensembles in which the diversification is achieved through diverse activation functions [76], ensembles obtained through negative correlation learning [110, 111, 112], bagging ensembles [72, 113], boosting ensembles [114, 115] deep ensembles [73, 74, 116] including deep echo-state network ensembles [117], and anchored ensembles [75] with a later modification random activation function (RAF) ensembles [76]. RAF ensembles take over the principle of anchored ensembles that regularization is performed not with respect to zero, but with respect to the initialization values of the parameters, which are assumed normally distributed. Differently to an anchored ensemble, however, an RAF ensemble uses varied ativation fuctions from an a priori specified set of size n_{AF}. From that set, the activation function is chosen randomly, apart from the first n_{AF} members of the ensemble, among which each activation function occurs exactly once. We consider this last mentioned kind of ensembles as the state of the art.

To our knowledge, the only ANNs estimating the distribution of function values that have already been used as surrogate models in BBO, are prior networks. In [71], the prediction accuracy of four versions has been evaluated on the above mentioned dataset from previous runs of DTS-CMA-ES. This direction of research is continued by the present paper: Section 4 reports results for CMA-ES surrogate-assisted by two variants of RAF ensembles.

3.3. ANNs Found Useful in Bayesian Optimization

Recall from Section 2 that GPs, simultaneously with their importance as surrogate models in BBO with non-Bayesian methods, such as CMA-ES, also play a crucial role in Bayesian optimization. That is why this subsection lists three well-known kinds of modern neural networks that have been recently found useful in Bayesian optimization. In our opinion, this indicates that they are worth investigating whether they could be used also in surrogate-assisted BBO.

1. *Variational autoencoders* have been utilized in Bayesian optimization because they allow for optimization in a lower-dimensional latent space [77, 78].
2. The *generative adversarial networks* (GANs) paradigm has been recently shown to be applicable to BBO: A generator proposes samples that align with the distribution of low values (or even the optimal value) of the black-box function, while one or more discriminators classify samples based on whether they belong to that distribution [79, 80].
3. *Transformers* have proven effective in estimating complex prior distributions for Bayesian optimization [81, 82]. Notably, an OptFormer transformer trained on Google Vizier [118], the largest hyperparameter optimization (HPO) database, achieved superior HPO outcomes compared to GP-based Bayesian optimization [81]. Furthermore, the recently introduced transformer-based Prior-data Fitted Networks [82] can mimic Gaussian Processes (GPs) and Bayesian networks, while also incorporating additional information into the prior.

3.4. ANN-Based Transfer Learning for Surrogate-Assisted Black-Box Optimization

Obtaining accurate surrogate models in the initial stages of BBO is challenging due to the scarcity of data points with evaluated objective function values. That can be mitigated by leveraging knowledge-transfer learning. And a connection of modern kinds of neural networks with transfer learning is even more obvious than with active learning. Indeed, transfer learning is nowadays one of the areas where ANNs play most important role [119, 120, 121]. Different types of ANNs have been utilized to this end, including convolutional [122, 123], recurrent [124], autoencoder [125, 126], GAN [127, 128, 129], and transformer [81]. In the context of the research direction pursued in this paper, most interesting are those that also have connections to BBO:

(i) Four ANN-based transfer learning approaches draw inspiration from the GAN paradigm. CoGAN trains two GANs to generate the source and target, respectively, achieves a domain invariant feature space by tying the high layers parameters of the two GANs, and performs domain adaptation by training a classifier on the discriminator output [130]. Adversarial discriminative domain adaptation learns first a discriminative representation using the labels in the source domain, and then, using a domain-adversarial loss, a separate encoding that maps the target data to the same space through an asymmetric mapping [127]. Minimax-game-based selective transfer learning employs a selector and a discriminator to identify source domain data resembling the target domain's distribution, and distinguish genuine target domain data from selected source domain data, respectively [129]. Selective adversarial network addresses negative transfer by excluding outlier classes from the source domain selection, and maximizing the similarity between source and target domain data distributions [128].

(ii) An autoencoder for transfer learning, described in [125, 126], incorporates embedding and label encoding layers. The embedding layer reduces the disparity between instance distributions from the source and target domains, while the label encoding layer utilizes a softmax regression model to encode label information from the source domain.

(iii) The transformer OptFormer has demonstrated competitiveness with specific transfer learning methods, although its usage leans more toward metalearning than traditional transfer learning [81].

4. Experimental Evaluation of RAF Ensembles

This section describes a small experimental contribution to one of the above surveyed possible research directions: RAF ensembles are experimentally evaluated as surrogate models for CMA-ES. The experiments were performed on the probably most commonly used platform for experimenting in continuous optimization – COCO (Comparing Continuous Optimizers) [92]. COCO contains severeal suites of benchmark functions, our evaluation was performed with the most traditional suite, which is the *bbob suite* [92]. It consists of 24 dimension-scalable noiseless benchmark functions, the definitions of which have been given in [91]. Each function is used in 15 differently rotated and/or translated instances. The employed benchmarks forming the bobo suite are surveyed in Appendix A.

4.1. Considered Variants of RAF Ensembles

As activation functions forming an RAF ensemble, we employed those included in the implementation [131], to which the RAF paper refers [76]. They are listed in Appendix B. We used them in two variants of RAF ensembles:

1. An RAF ensemble of size 5 trained directly using the above mentioned implementation [131], and aggregated by the empirical mean. In the results, it will be denoted simply *RAF*.
2. An ensemble of size 5, in which the differences of values of the original black-box objective function with respect to its median are first transformed to their logarithms before using [131] in the logarithmic scale to train the ensemble. This transformation attempts to deal with situations when the function returns in many points values close to the median. The aggregation function is again the empirical mean, which in terms of the data before the logarithmic transformation actually corresponds to the empirical geometric mean. That version will be in the results denoted *RAF-log*.

4.2. Considered CMA-ES Variants for Comparison

CMA-ES surrogate-assisted by the above mentioned two variants of RAF ensembles was compared with CMA-ES without surrogate modelling, as well as with two earlier surrogate-assisted variants of CMA-ES:

3. CMA-ES without surrogate modelling was used in an implementation that is in the COCO data archive [132] called default-CMA-ES, and described as "default CMA-ES from the pycma module, version 3.3.0". Here, it will be in the results denoted simply *default*.
4. DTS-CMA-ES [12], using a surrogate GP with the covariance function Matérn $\frac{5}{2}$. In the results, it will be denoted simply *DTS*.
5. lq-CMA-ES [20], which will be in the results denoted simply *lq*.

4.3. Evolution Control

Whereas DTS-CMA-ES and lq-CMA-ES have each their own evolution control, for the two variants of RAF ensembles was necessary to propose when to evaluate a given point x by the original black-box objective function F_{bb}, and when by its surrogate model F_{sm}. We decided to use a modification of the lq-CMA-ES evolution control. That modification is described below in Algorithm 1 using the notation $\tau((y_1,\dots,y_k),(z_1,\dots,z_k))$ for the Kendall correlation coefficient between the sequences $(y_1,\dots,y_k)$ and $(z_1,\dots,z_k)$, and the notation ρ for the ranking function on $\mathbb{R}^d$, i.e.,

$$\rho:\mathbb{R}^d\to\Pi(d) \text{ with } \Pi(d) \text{ denoting the set of permutations of } \{1,\dots,d\}$$
$$\text{such that } \forall y\in\mathbb{R}^d: (\rho(y))_i<(\rho(y))_j \Rightarrow y_i\le y_j. \quad (1)$$

4.4. Results

In Tables 2–3, the two considered variants of RAF ensembles, and three considered other CMA-ES variants, are compared based on the difference between the optimal value of the objective function, and its value achieved for a given evaluation budget. The achieved values were averaged over the 15 instances provided by the COCO benchmark suite in each dimension for each of the 24 noiseless functions listed in Appendix A. The comparisons were performed separately for each of the five above described groups of those functions, and subsequently also for all 24 of them, each time including the instances in dimensions 2, 3, 5, 10, and 20. For each evaluation budget, hence, six evaluations were

Table 2
Comparison of CMA-ES surrogate-assisted by RAF, and by RAF-log, with CMA-ES without surrogate modelling, with lq-CMA-ES, and with DTS-CMA-ES, for evaluation budget 3×dimension. Each cell of each sub-table records the number of function-dimension combinations, for which the method in the row achieved with the evaluation budget a lower value, averaged over the 15 COCO instances, than the method in the column. Ties within the considered precision are halved between both methods. If the Friedman test rejected the hypothesis of equivalence of all methods, and according to the subsequent Wilcoxon signed-rank test with Holm correction, the method in the row is significantly better than the method in the column, the number in the cell is in bold with * for the familywise level 5 %, and with ** for the familywise level 1 %.

Separable functions

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	10.5	7	0	13.5
RAF-log	14.5	-	10	0.5	13.5
DTS-CMA-ES	18	15	-	1	17.5
lq-CMA-ES	**25****	**24.5****	**24****	-	**23****
default CMA-ES	11.5	11.5	7.5	2	-

Functions with low or moderate conditioning

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	10	3.5	2	6.5
RAF-log	10	-	4.5	3.5	7
DTS-CMA-ES	**16.5***	15.5	-	8	**15***
lq-CMA-ES	**18****	16.5	12	-	**16****
default CMA-ES	13.5	13	5	4	-

Unimodal functions with high conditioning

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	12.5	4	2	10.5
RAF-log	12.5	-	9.5	5	12
DTS-CMA-ES	**21***	15.5	-	10	16
lq-CMA-ES	**23****	**20***	15	-	18.5
default CMA-ES	14.5	13	9	6.5	-

Multi-modal functions with adequate global structure

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	16	9.5	9	10
RAF-log	9	-	8.5	8.5	6.5
DTS-CMA-ES	15.5	16.5	-	14.5	15
lq-CMA-ES	16	17	10.5	-	13
default CMA-ES	15	18.5	10	12	-

Multi-modal functions with weak global structure

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	11	5.5	5	6.5
RAF-log	14	-	8.5	5	6
DTS-CMA-ES	19.5	16.5	-	13.5	15
lq-CMA-ES	**20***	20	11.5	-	15.5
default CMA-ES	18.5	18	10	9.5	-

All noiseless benchmark functions

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	60	29.5	18	47
RAF-log	60	-	41	22	45
DTS-CMA-ES	**90.5****	**79****	-	47	**78.5****
lq-CMA-ES	**102****	**98****	73	-	**86****
default CMA-ES	73	75	41.5	34	-

Table 3

Comparison of CMA-ES surrogate-assisted by RAF, and by RAF-log, with CMA-ES without surrogate modelling, with lq-CMA-ES, and with DTS-CMA-ES, for evaluation budget $50\times$dimension. Each cell of each sub-table records the number of function-dimension combinations, for which the method in the row achieved with the evaluation budget a lower value, averaged over the 15 COCO instances, than the method in the column. Ties within the considered precision are halved between both methods. If the Friedman test rejected the hypothesis of equivalence of all methods, and according to the subsequent Wilcoxon signed-rank test with Holm correction, the method in the row is significantly better than the method in the column, the number in the cell is in bold with * for the familywise level 5 %, and with ** for the familywise level 1 %.

Separable functions

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	7	4	1.5	11
RAF-log	18	-	6	0.5	12
DTS-CMA-ES	**21****	19	-	7.5	19
lq-CMA-ES	**23.5****	**24.5****	17.5	-	**22.5****
default CMA-ES	14	13	6	2.5	-

Functions with low or moderate conditioning

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	15.5	6	0.5	4.5
RAF-log	4.5	-	6	0	2
DTS-CMA-ES	14	14	-	11.5	14
lq-CMA-ES	**19.5****	**20****	8.5	-	**18.5***
default CMA-ES	15.5	**18****	6	1.5	-

Unimodal functions with high conditioning

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	**21***	0	0	4
RAF-log	4	-	0	0	1
DTS-CMA-ES	**25****	**25****	-	7.5	**25****
lq-CMA-ES	**25****	**25****	**17.5***	-	**25****
default CMA-ES	21	**24****	0	0	-

Multi-modal functions with adequate global structure

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	CMA-ES alone
RAF	-	15	13	10	11.5
RAF-log	10	-	11.5	9	12
DTS-CMA-ES	12	13.5	-	13.5	15
lq-CMA-ES	15	16	11.5	-	13
default CMA-ES	13.5	13	10	12	-

Multi-modal functions with weak global structure

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	9	3.5	8	14
RAF-log	16	-	6.5	12	18.5
DTS-CMA-ES	21.5	18.5	-	20	**23****
lq-CMA-ES	17	13	5	-	**20***
default CMA-ES	11	6.5	2	5	-

All noiseless benchmark functions

	RAF	RAF-log	DTS-CMA-ES	lq-CMA-ES	default CMA-ES
RAF	-	67.5	26.5	20	45
RAF-log	52.5	-	30	21.5	45.5
DTS-CMA-ES	**93.5****	**90****	-	60	**96****
lq-CMA-ES	**100****	**98.5****	60	-	**99****
default CMA-ES	75	74.5	24	21	-

Algorithm 1 Evolution control used for RAF and RAF-log ensembles.

```
Require: points $x_1, \ldots, x_\lambda \in \mathbb{R}^d$, in which the surrogate model $F_{\text{sm}}$ trained on some archive $A$ has
    been evaluated; thus $\lambda$ is the population size
Set $k = \lfloor 1 + \max(0.02\lambda, 4) \rfloor$; the number of $F_{\text{bb}}$ evaluations
Set $Q = \{x_j | (\rho(F_{\text{sm}}(x_1), \ldots, F_{\text{sm}}(x_\lambda)))_j \leq k\}$; points with the $k$ smallest $F_{\text{sm}}$ values
In $x \in Q$ for which $F_{\text{bb}}(x)$ is not yet known, evaluate $F_{\text{bb}}(x)$
Order the elements of $Q$ as $(x_Q^1, \ldots, x_Q^k)$ decreasingly with respect to their $F_{\text{bb}}(x)$ values
Set $\ell = \max(1, \lfloor k + 1 - \max(15, 0.75\lambda) \rfloor)$; the lower index for computing $\tau$ between $F_{\text{bb}}$ and $F_{\text{sm}}$
while $k < \lambda$ & $\tau((F_{\text{bb}}(x_\ell), \ldots, F_{\text{bb}}(x_k)), (F_{\text{sm}}(x_\ell), \ldots, F_{\text{sm}}(x_k))) < 0.85$ do
    Update $Q = Q \cup \{x_j | (\rho(F_{\text{sm}}(x_1), \ldots, F_{\text{sm}}(x_\lambda)))_j \leq \lceil 1.5k \rceil\}$
    In $x \in Q$ for which $F_{\text{bb}}(x)$ is not yet known, evaluate $F_{\text{bb}}(x)$
    Update $k = \lceil 1.5k \rceil, \ell = \max(1, \lfloor k + 1 - \max(15, 0.75\lambda) \rfloor)$
    Order the elements of $Q$ as $(x_Q^1, \ldots, x_Q^k)$ decreasingly with respect to their $F_{\text{bb}}(x)$ values
end while
Update $A = A \cup \{x_j | F_{\text{bb}}(x)$ has been evaluated in $x_j\}$
if $k = \lambda$ then
    Return $A$ and $\{(x_1, F_{\text{bb}}(x_1)), \ldots, (x_\lambda), F_{\text{bb}}(x_\lambda))\}$
else
    Return $A$ and $\{(x_i, F_{\text{bb}}(x_i)) | x_i \in Q\} \cup \{(x_i, F_{\text{sm}}(x_i)) | i = 1, \ldots, \lambda, x_i \notin Q\}$
end if
```

performed. The comparisons in Table 2 were conducted for the evaluation budget $3\times$dimension, while the comparisons in Table 3 were conducted for the evaluation budget $50\times$dimension.

The results of each of those 12 comparisons were subsequently assessed for statistical significance. First, the hypothesis that all five considered methods are equivalent was tested by the Friedman test. With the exception of both comparisons for multi-modal functions with adequate global structure, the test rejected that hypothesis on the familywise significance level 5%, using the Holm procedure for multiple-hypothesis correction [133]. This rejection justified testing the equivalence of any two among the five methods. We adopted the arguments of [134] that, in machine learning, the Wilcoxon signed-rank test is more appropriate for this purpose than the post-hoc tests presented in [135] and [133]. If for particular two methods, the Wilcoxon signed-rank test rejected the hypothesis that they are equivalent, then in the respective table, their comparison in the row corresponding to the method that was more frequently better is shown in bold italics.

The results in Tables 2–3 primarily confirm the superior performance of the methods lq-CMA-ES, and DTS-CM-ES. In the two comparisons based on all 120 noiseless benchmark functions, each of them is for both considered budgets significantly better not only than default CMA-ES, but also than CMA-ES surrogate-assisted by the two variants of RAF ensembles. Moreover, lq-CMA-ES is also among the 10 comparisons based on individual groups of functions 6 times significantly better than default CMA-ES, and 7 times, respectively 5 times significantly better than CMA-ES surrogate-assisted by RAF, respectively by RAF-log. For DTS-CMA-ES, the results of the 10 comparisons based on individual groups of functions are less convincing: 3 times significantly better than default CMA-ES, 3 times than CMA-ES surrogate-assisted by RAF, and only once than CMA-ES assisted by RAF-log. As to a comparison between the two variants of RAF ensembles, the differences among them were not significant apart from unimodal functions with high conditioning, for which CMA-ES achieves significantly better results if assisted by RAF than if assisted by RAF-log.

The different progress of optimization performed by each of the compared methods is illustrated, always in three particular dimensions, by means of optimization-progress plots. They show the average difference Δ_f between the optimal and achieved value of the objective function over the 15 COCO instances. For that illustration, we have chosen the functions f_9 (Figure 1), f_{18} (Figure 2), and f_{20} (Figure 3). We can see that optimisation using CMA-ES surrogate-assisted by RAF or RAF-log sometimes leads to similarly fast decrease of the objective function as, or even faster than, optimization using

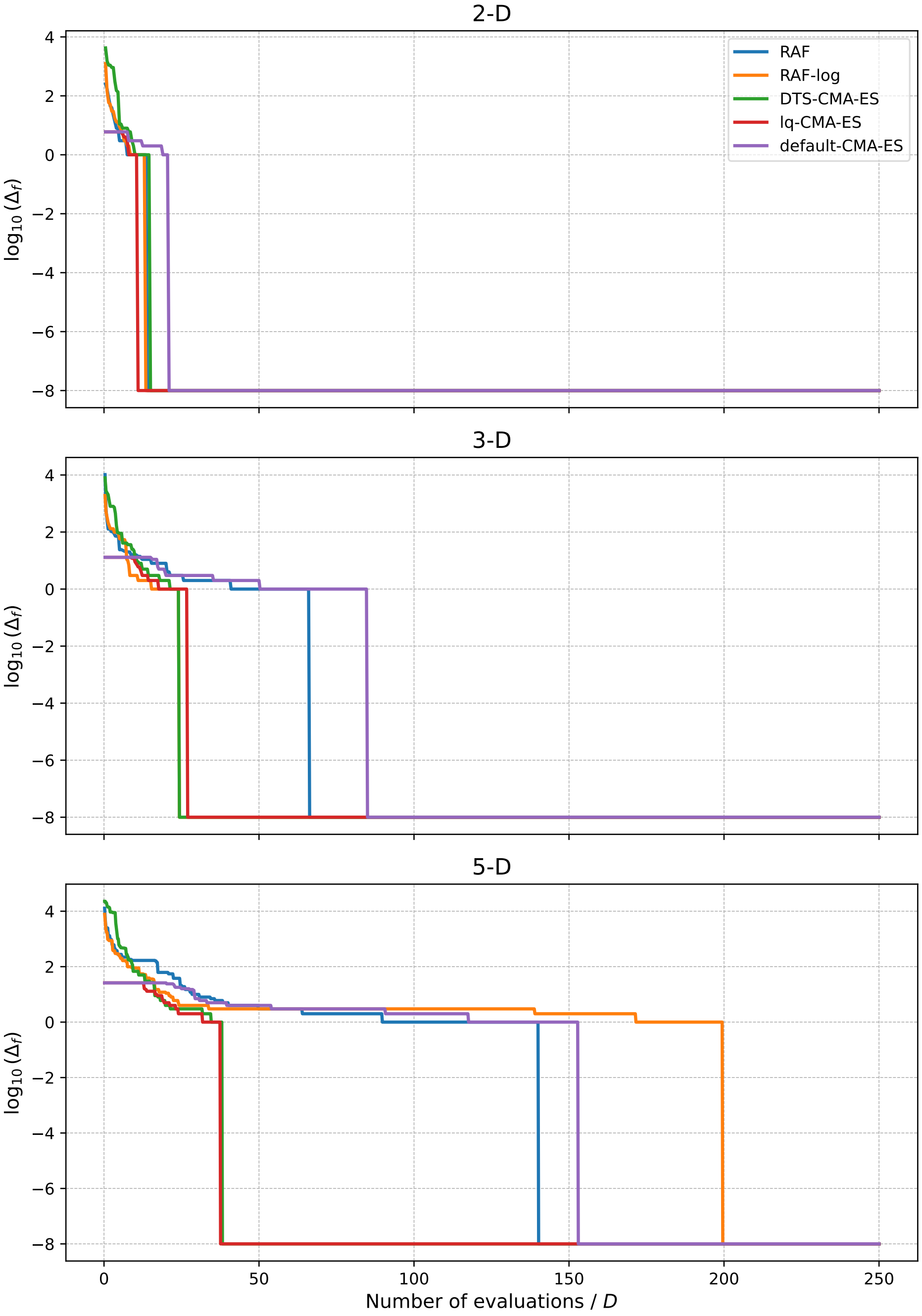

Figure 1: Progress of optimization by the compared methods up to the budget 250×dimension for the benchmark function f_9 – Rosenbrock rotated. Each curve is the average of the 15 COCO instances of this function.

the state-of-the-art methods DTS-CMA-ES or lq-CMA-ES. In Figure 1, this is the case for RAF-log in dimension 2. In Figure 2, dimesnion 3, CMA-ES surrogate-assisted by RAF reaches lower values of the objective function than any other of the compared methods, whereas in dimension 2, CMA-ES surrogate-assisted by any of RAF or RAF-log leads to a similarly fast decrease of f_{18} as DTS-CMA-ES but slower than lq-CMA-ES. Finally, in Figure 3, dimensions 3 and 5, CMA-ES surrogate-assisted by any of RAF or RAF-log leads to a similarly fast decrease of f_{18} as lq-CMA-ES, but slower than DTS-CMA-ES.

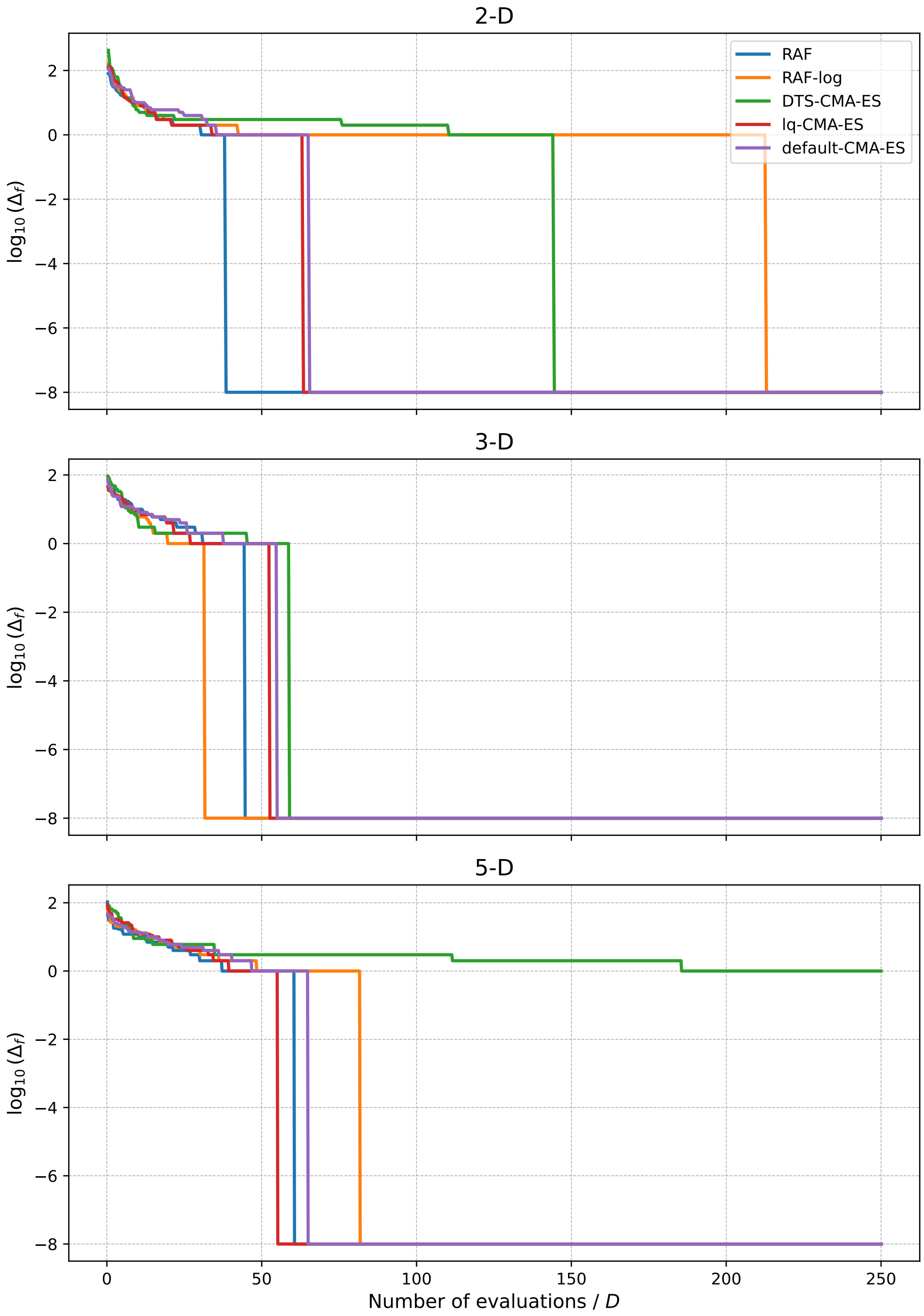

Figure 2: Progress of optimization by the compared methods up to the budget $250\times$dimension for the benchmark function f_{18} – Schaffers F7 function, moderately ill-conditioned. Each curve is the average of the 15 COCO instances of this function.

5. Conclusion

The paper was motivated by our opinion that the intense and successful development of artificial neural networks during the last 15 years suggests that they again have the potential to be important for active learning in surrogate-assisted BBO. It surveyed possible directions of research into that potential, including closely connected research into neural-network-based transfer learning for surrogate modelling. Moreover, it recalled the first published investigations in some of those directions, and added a new contribution to the emerging mosaic of those investigations.

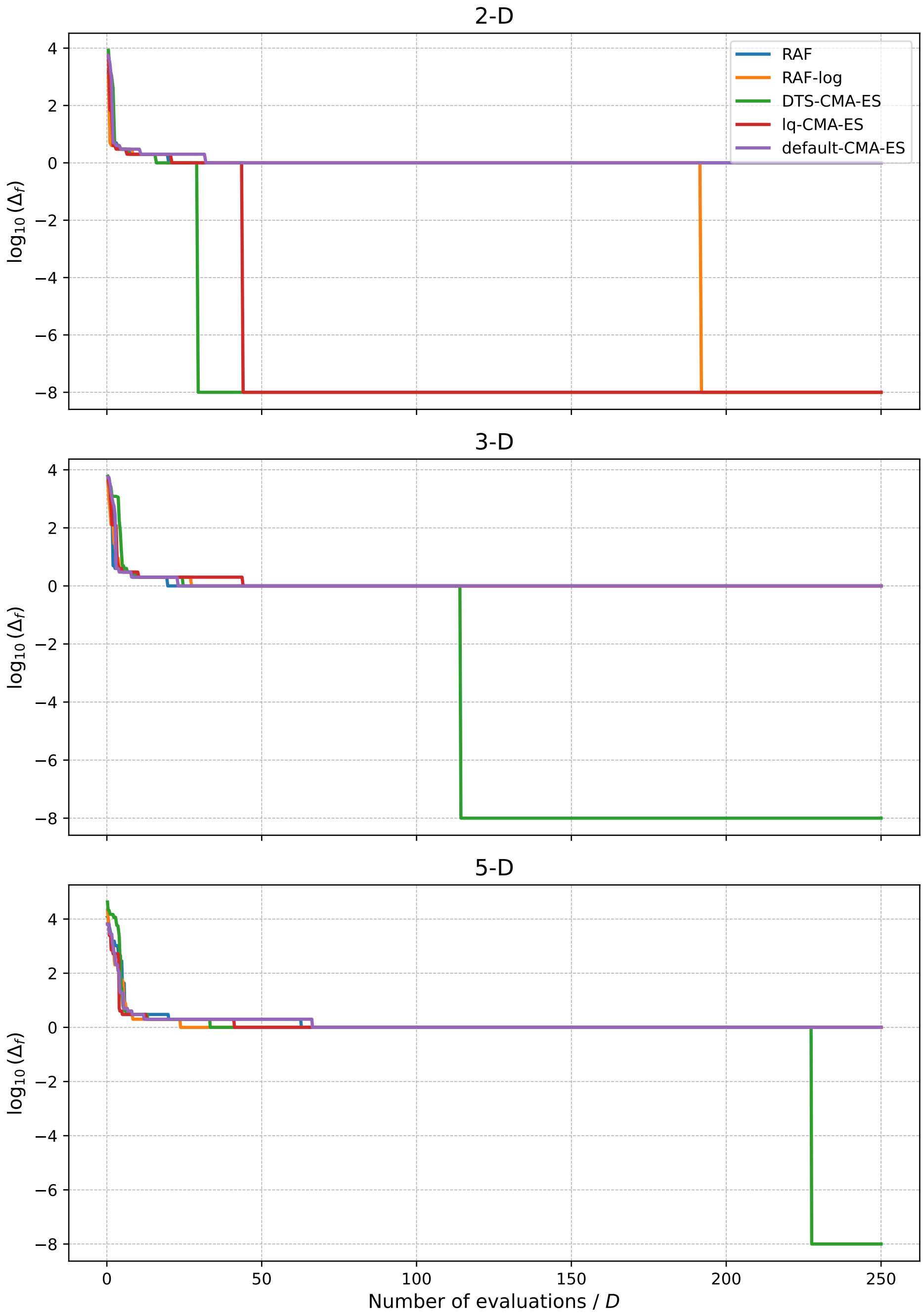

Figure 3: Progress of optimization by the compared methods up to the budget 250×dimension for the benchmark function f_{20} – Schwefel. Each curve is the average of the 15 COCO instances of this function.

The fact that the main purpose of the experimental section of the paper is to contribute to the mosaic of emerging investigations should be epmhasized especially in context of the obtained experimental results. It justifiess that there is no significant difference between using CMA-ES surrogate-assisted by RAF ensembles and using it alone, as well as that results with RAF-ensemble-based surrogate models are significantly worse than results with the state-of-the-art surrogate-assisted CMA-ES variants, lq-CMA-ES, and DTS-CMA-ES. This is an obvious limitation not only of RAF ensembles, but of all above surveyed kinds of neural networks that have been so far investigated as surrogate models for CMA-ES. On the other hand, as the survey has shown, there are many more other possibilities for such investigations within future research.

Acknowledgemengt

The research reported in this paper has been supported by the German Research Foundation (DFG) funded project 467401796, and by the Czech Technical University grant SGS 23/205/OHK3/3T/18. The authors are very grateful to Jaroslav Langer for his crucial contribution to the RAF experiments.

References

[1] M. Baerns, M. Holeňa, Combinatorial Development of Solid Catalytic Materials. Design of High-Throughput Experiments, Data Analysis, Data Mining, Imperial College Press / World Scientific, London, 2009.

[2] A. Booker, J. Dennis, P. Frank, D. Serafini, T. V., M. Trosset, A rigorous framework for optimization by surrogates, Structural and Multidisciplinary Optimization 17 (1999) 1–13.

[3] M. El-Beltagy, P. Nair, A. Keane, Metamodeling techniques for evolutionary optimization of computaitonally expensive problems: Promises and limitations, in: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, 1999, pp. 196–203.

[4] A. Ratle, Kriging as a surrogate fitness landscape in evolutionary optimization, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 15 (2001) 37–49.

[5] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, K. Giannakoglou, Metamodel-assisted evolution strategies, in: PPSN, ACM, 2002, pp. 361–370.

[6] S. Leary, A. Bhaskar, A. Keane, A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation, Journal of Global Optimization 30 (2004) 39–58.

[7] Y. Ong, P. Nair, A. Keane, K. Wong, Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems, in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation, Springer, 2005, pp. 307–331.

[8] K. Rasheed, X. Ni, S. Vattam, Methods for using surrogate modesl to speed up genetic algorithm oprimization: Informed operators and genetic engineering, in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation, Springer, 2005, pp. 103–123.

[9] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Transactions on Evolutionary Computation 6 (2002) 481–494.

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

[11] C. Huang, B. Radi, A. El Hami, H. Bai, CMA evolution strategy assisted by kriging model and approximate ranking, Applied Intelligence 48 (2018) 4288–4204.

[12] L. Bajer, Z. Pitra, J. Repický, M. Holeňa, Gaussian process surrogate models for the CMA evolution strategy, Evolutionary Computation 27 (2019) 665–697.

[13] Z. Pitra, M. Hanuš, J. Koza, J. Tumpach, M. Holeňa, Interaction between model and its evolution control in surrogate-assisted CMA evolution strategy, in: GECCO, 2021, p. 358 (paper no.).

[14] P. Dufossé, N. Hansen, Augmented Lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES, in: GECCO, 2021, pp. 519–527.

[15] N. Sakamoto, Y. Akimoto, Adaptive ranking-based constraint handling for explicitly constrained black-box optimization, Evolutionary Computation 30 (2022) 503–529.

[16] I. Loshchilov, M. Schoenauer, M. Sebag, A mono surrogate for objective optimization, in: GECCO, 2010, pp. 471–478.

[17] F. Gibson, R. Everson, J. Fieldsend, Guiding surrogate-assisted multi-objective optimisation with decision maker preferences, in: GECCO, 2022, pp. 786–795.

[18] S. Kern, N. Hansen, P. Koumoutsakos, Local metamodels for optimization using evolution strategies, in: PPSN, 2006, pp. 939–948.

[19] A. Auger, D. Brockhoff, N. Hansen, Benchmarking the local metamodel cma-es on the noiseless BBOB'2013 test bed, in: GECCO, 2013, pp. 1225–1232.

[20] N. Hansen, A global surrogate assisted CMA-ES, in: GECCO, 2019, pp. 664–672.

[21] H. Yu, C. Sun, Y. Tan, J. Zeng, Y. Jin, An adaptive model selection strategy for surrogate- assisted particle swarm optimization algorithm, in: IEEE SCI, 2016, pp. 1–8.

[22] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Transactions on Cybernetics 47 (2017) 2664–2677.

[23] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Structural optimization using evolution strategies and neural networks, Computer Methods in Applied Mechanics and Engineering 156 (1998) 309–333.

[24] H. Ulmer, F. Streichert, A. Zell, Model-assisted steady state evolution strategies, in: GECCO, Springer, 2003, pp. 610–621.

[25] D. Lim, Y. Ong, Y. Jin, S. B., A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation, in: GECCO, 2007, pp. 1288–1295.

[26] L. Bajer, M. Holeňa, Surrogate model for continuous and discrete genetic optimization based on RBF networks, in: Intelligent Data Engineering and Automated Learning, Springer, 2010, pp. 251–258.

[27] L. Na, Q. Feng, Z. Liang, W. Zhong, Gaussian process assisted coevolutionary estimation of distribution algorithm for computationally expensive problems, Journal of Central South University of Technology 19 (2012) 443–452.

[28] V. Volz, G. Rudolph, B. Naujoks, Investigating uncertainty propagation in surrogate-assisted evolutionary algorithms, in: GECCO, 2017, pp. 881–888.

[29] L. Toal, D. Arnold, Simple surrogate model assisted optimization with covariance matrix adaptation, in: PPSN, 2020, pp. 184–197.

[30] Z. Li, T. Gao, B. Wang, Elite-driven surrogate assisted CMA-ES algorithm by improved lower confidence bound method, in: Engineering with Computers, Springer, 2022, pp. 10.1007/s00366–022–01642–5 (doi).

[31] Z. Zhou, Y. Ong, P. Nair, A. Keane, K. Lum, Combining global and local surrogate models to accellerate evolutionary optimization, IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews 37 (2007) 66–76.

[32] B. Saini, M. Lópey-Ibañez, K. Miettinen, Automatic surrogate modelling technique selection based on features of optimization problems, in: GECCO, 2019, pp. 1765–1772.

[33] N. Belkhir, J. Dréo, P. Savéant, M. Schoenauer, Per instance algorithm configuration of CMA-ES with limited budget, in: GECCO, 2017, pp. 681–688.

[34] Z. Pitra, J. Repický, M. Holeňa, Boosted regression forest for the doubly trained surrogate covariance matrix adaptation evolution strategy, in: ITAT 2018, 2018, pp. 72–79.

[35] T. Runarsson, Ordinal regression in evolutionary computation, in: PPSN, 2006, pp. 1048–1057.

[36] I. Loshchilov, M. Schoenauer, M. Sebag, Comparison-based optimizers need comparison-based surrogates, in: PPSN, 2010, pp. 364–373.

[37] A. A., D. Arnold, Adaptive function value warping for surrogate model assisted evolutionary optimization, in: PPSN, 2022, pp. 76–89.

[38] I. Couckuyt, D. Gorissen, Automatic surrogate model type selection during the optimization of expensive black-box problems, in: Winter Simulation Conference, 2011, pp. 4285–4293.

[39] H. Dong, S. Sun, B. Song, P. Wang, Multi-surrogate-based global optimization using a score-based infill criterion, Structural and Multidisciplinary Optimization 59 (2019) 485–506.

[40] N. Hansen, A. Ostermaier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 9 (2001) 159–195.

[41] N. Hansen, The CMA evolution strategy: A comparing review, in: Towards a New Evolutionary Computation, Springer, 2006, pp. 75–102.

[42] M. Wu, A. Karkar, B. Liu, A. Yakovlev, G. Gielen, V. Grout, Network on chip optimization based on surrogate model assisted evolutionary algorithms, in: IEEE CEC, 2014, pp. 3266–3271.

[43] D. Jones, M. Schonlau, W. Welch, Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13 (1998) 455–492.

[44] J. Knowles, ParEGO: a hybrid algorithm with on-line landscape approximation for expensive

multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 10 (2006) 50–66.

[45] Y. Diouane, V. Picheny, R. Le Riche, A. Di Perrotolo, TREGO: a trust-region framework for efficient global optimization, Journal of Global Optimization 85 (2022) 10.1007/s10898–022–01245–w (doi).

[46] H. Mohammadi, R. Riche, E. Touboul, Making EGO and CMA-ES complementary for global optimization, in: Learning and Intelligent Optimization, Springer, 2015, pp. 287–292.

[47] Z. Pitra, L. Bajer, M. Holeňa, Doubly trained evolution control for the surrogate cma-es, in: PPSN, 2016, pp. 59–68.

[48] Y. He, Y. Yuen, Black box algorithm selection by convolutional neural network, in: LOD, 2020, pp. 264–280.

[49] M. Pikalov, V. Mironovich, Automated parameter choice with exploratory landscape analysis and machine learning, in: GECCO, 2021, pp. 1982–1985.

[50] R. Prager, M. Seiler, H. Trautman, P. Kerschke, Towards feature-free automated algorithm selection for single-objective continuous black box optimization, in: IEEE SCI, 2021, pp. 1–8.

[51] Z. Pitra, L. Bajer, M. Holeňa, Knowledge-based selection of gaussian process surrogates, in: ECML Workshop IAL, 2019, pp. 48–63.

[52] Z. Pitra, J. Repický, M. Holeňa, Landscape analysis of Gaussian process surrogates for the covariance matrix adaptation evolution strategy, in: GECCO, ACM, 2019, pp. 691–699.

[53] R. Seiler, M.V.and Prager, P. Kerschke, H. Trautmann, A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes, in: GECCO, 2022, pp. 657–665.

[54] A. Jankovic, G. Popovski, T. Eftimov, C. Doerr, The impact of hyper-parameter tuning for landscape-aware performance regression and algorithm selection, in: GECCO, 2021, pp. 687–696.

[55] R. Calandra, J. Peters, C. Rasmussen, M. Deisenroth, Manifold Gaussian processes for regression, in: IJCNN, 2016, pp. 3338–3345.

[56] A. Wilson, Z. Hu, R. Salakhutdinov, E. Xing, Deep kernel learning, in: ICAIS, 2016, pp. 370–378.

[57] J. Koza, J. Tumpach, Z. Pitra, M. Holeňa, Combining gaussian processes and neural networks in surrogate modeling for covariance matrix adaptation evolution strategy, in: IAL Workshop, ECML PKDD, 2021, pp. 1–10.

[58] J. Ružička, J. Koza, J. Tumpach, Z. Pitra, M. Holeňa, Combining gaussian processes with neural networks for active learning in optimization, in: ECML Workshop IAL, 2021, pp. 105–120.

[59] H. Salimbeni, M. Deisnroth, Doubly stochastic variational inference for deep Gaussian processes, in: NeurIPS, 2017, pp. 1–16.

[60] K. Blomqvist, S. Kaski, M. Heinonen, Deep convolutional Gaussian processes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2020, pp. 582–597.

[61] G. Hernández-Muñoz, C. Villacampa-Calvo, D. Hernández Lobato, Deep Gaussian processes using expectation propagation and Monte Carlo methods, in: ECML PKDD, 2020, pp. 479–494.

[62] D. Ming, D. Williamson, S. Guillas, Deep Gaussian process emulation using stochastic imputation, Techometrics 65 (2022) 150–161.

[63] A. Sauer, R. Gramacy, D. Hugdon, Active learning for deep gaussian process surrogates, Technometrics 65 (2023) 4–18.

[64] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, in: NeurIPS, 2018, pp. 1–10.

[65] R. Novak, L. Xiao, J. Hron, J. Lee, A. Alemi, et al., Neural tangents: Fast and easy infinite neural networks in python, in: ICLR, 2020, pp. 1–19.

[66] A. Malinin, M. Gales, Predictive uncertainty estimation via prior networks, in: NeurIPS, 2018, pp. 1–17.

[67] M. Biloš, B. Charpentier, S. Günnemann, Uncertainty on asynchronous time event prediction, in: NeurIPS, 2019, pp. 1–10.

[68] A. Malinin, M. Gales, Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness, in: NeurIPS, 2019, pp. 1–12.

[69] J. Nandy, W. Hsu, M. Lee, Towards maximizing the representation gap between in-domain and

out-of-distribution examples, in: NeurIPS, 2020, pp. 1–12.

[70] X. Zhao, F. Chen, S. Hu, J. Cho, Uncertainty aware semi-supervised learning on graph data, in: NeurIPS, 2020, pp. 1–10.

[71] J. Tumpach, J. Koza, Z. Pitra, M. Holeňa, Neural-network-based estimation of normal distributions in black-box optimization, in: ESANN, 2022, pp. 1–6.

[72] C. Valle, F. Saravia, H. Allende, R. Monge, C. Fernández, Parallel approach for ensemble learning with locally coupled neural networks, Neural Processing Letters 32 (2010) 277–291.

[73] B. Lakshminaraynan, A. Prityel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: NeurIPS, 2017, pp. 1–12.

[74] R. Hu, Q. Huang, S. Chang, H. Wang, J. He, The MBPEP: A deep ensemble pruning algorithm providing high quality uncertainty prediction, Applied Intelligence 49 (2019) 2942–2955.

[75] T. Pearce, F. Leibfried, A. Brintrup, M. Zaki, A. Neely, Uncertainty in neural networks: Approximately bayesian ensembling, in: AISTATS, 2020, pp. 1–30.

[76] Y. Stoyanova, S. Ghandi, M. Tavakol, Toward robust uncertainty estimation with random activation functions, in: AAAI Conference on Artificial Intelligence, 2023, pp. 1–13.

[77] S. Kim, P. Lu, C. Lob, J. Smith, J. Snoek, et al., Deep learning for bayesian optimization of scientific problems with high-dimensional structure, Transactions on Machine Learning Research 1 (2022) openreview tPMQ6Je2rB.

[78] A. Tripp, E. Daxberger, J. Hernández-Lobato, Sample-efficient optimization in the latent space of deep generative models viaweighted retraining, in: NeurIPS, 2020, pp. 1–14.

[79] M. Gillhofer, H. Ramsauer, J. Brandstetter, B. Schäfl, S. Hochreiter, A GAN based solver of black-box inverse problems, in: NeurIPS, 2019, pp. 1–5.

[80] M. Lu, S. Ning, S. Liu, F. Sun, B. Zhang, et al., OPT-GAN: A broad-spectrum global optimizer for black-box problems by learning distribution, 2022. Arxiv 2102.03888v5.

[81] Y. Chen, X. Song, C. Lee, Z. Wang, Q. Zhang, et al., Towards learning universal hyperparameter optimizers with transformers, in: NeurIPS, 2022, pp. 1–16.

[82] S. Muller, M. Feurer, N. Hollmann, F. Hutter, PFNs4BO: In-context learning for Bayesian optimization, in: ICML, 2023, pp. 1–27.

[83] E. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, 2006.

[84] A. Damianou, N. Lawrence, Deep Gaussian processes, in: AISTATS, 2013, pp. 1–9.

[85] T. Bui, D. Hernandez-Lobato, J. Hernandez-Lobato, Y. Li, R. Turner, Deep Gaussian processes for regression using approximate expectation propagation, in: ICML, 2016, pp. 1472–1481.

[86] K. Cutajar, E. Bonilla, P. Michiardi, M. Filippone, Random feature expansions for deep Gaussian processes, in: ICML, 2017, pp. 884–893.

[87] A. Hebbal, L. Brevault, M. Balesdent, E. Talbi, N. Melab, Efficient global optimization using deep Gaussian processes, in: IEEE CEC, 2018, pp. 1–12.

[88] A. Matthews, J. Hron, M. Rowland, R. Turner, Gaussian process behaviour in wide deep neural networks, in: ICLR, 2019, pp. 1–15.

[89] A. Hebbal, L. Brevault, M. Balesdent, E. Talbi, N. Mela, Bayesian optimization using deep Gaussian processes, 2019. ArXiv: 1905.03350v1.

[90] H. Yu, B. Low, P. Jaillet, D. Liu, Convolutional normalizing flows for deep Gaussian processes, in: IJCNN, 2021, pp. 1–5.

[91] N. Hansen, S. Finck, R. Ros, A. Auger, Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions, Technical Report, INRIA, Paris Saclay, 2009.

[92] N. Hansen, A. Auger, R. Ros, O. Merseman, T. Tušar, D. Brockhoff, COCO: a platform for comparing continuous optimizers in a black box setting, Optimization Methods and Software 35 (2021) 114–144.

[93] J. Koza, J. Tumpach, Z. Pitra, M. Holeňa, Using past experience for configuration of Gaussian processes in black-box optimization, in: LION, 2021, pp. 167–182.

[94] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, et al., Deep neural networks as Gaussian processes, in: ICLR, 2018, pp. 1–17.

[95] R. Novak, L. Xiao, J. Lee, Y. Bahri, et al., Bayesian deep convolutional networks with many channels are Gaussian processes, in: ICLR, 2019, pp. 1–35.

[96] B. He, B. Lakshminarayanan, Y. Teh, Bayesian deep ensembles via the neural tangent kernel, in: NeurIPS, 2020, pp. 1–13.

[97] B. Paria, B. Pòczos, K. Ravikumar, S. J., A. Suggala, et al., Be greedy -- a simple algorithm for blackbox optimization using neural networks, in: ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World, 2022, pp. 1–27.

[98] A. Malinin, S. Chervontsev, I. Povilkov, M. Gales, Regression prior networks, 2020. ArXiv: 2006.11590v2.

[99] A. Amini, W. Schwarting, A. Soleimany, D. Rus, Deep evidential regression, in: NeurIPS, 2020, pp. 1–11.

[100] M. Sensoy, L. Kaplan, M. Kandmir, Evidential deep learning to quantify classification uncertainty, in: NeurIPS, 2018, pp. 1–11.

[101] D. Oh, B. Shin, Improving evidential deep learning via multi-task learning, in: AAAI Conference on Artificial Intelligence, 2022, pp. 1–14.

[102] Y. Tong, P. Xu, T. Denoeux, An evidential classifier based on Dempster-Shafer theory and deep learning, Neurocomputing 450 (2021) 275–293.

[103] D. Ulmer, A survey on evidential deep learning for single-pass uncertainty estimation, 2021. ArXiv: 2110.03051v2.

[104] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[105] J. Ling, Z. Zhou, Causal discovery based on neural network ensemble method, Journal of Software 15 (2004) 1479–1484.

[106] H. Chen, S. Yuan, K. Jiang, Wrapper approach for learning neural network ensemble by feature selection, in: Advances in Neural Networks – ISNN 2005, Springer, 202, pp. 526–531.

[107] D. Partridge, Network generalization differences quantified, Neural Networks 9 (1996) 263–271.

[108] M. Jang, S. Cho, Observational learning algorithm for an ensemble of neural networks, Pattern Analysis and Applications 5 (2002) 154–167.

[109] Z. Wang, S. Chen, Z. Chen, An active learning approach for neural network ensemble, Journal of Computer Research and Development 42 (2005) 375–380.

[110] M. Islam, X. Yao, K. Murase, A constructive algorithm for training cooperative neural network ensembles, IEEE Transactions on Neural Networks 14 (2003) 820–834.

[111] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Information Sciences 264 (2014) 104–117.

[112] K. Dai, J. Zhao, F. Cao, A novel decorrelated neural network ensemble algorithm for face recognition, Knowledge Based Systems 89 (2015) 541–552.

[113] J. Ling, Z. Chen, Z. Zhou, Feature selection based neural network ensemble method, Journal of Fudan University (Natural Sciences) 43 (2004) 685–688.

[114] T. Yang, C. Zhang, Freeway incident detection based on Adaboost RBF neural network, Computer Engineering and Applications 32 (2008) 223–225.

[115] H. Liu, G. Chen, G. Song, T. Han, AdaBoost based ensemble of neural networks in analog circuit fault diagnosis, Chinese Journal of Scientific Instrument 4 (2010) 851–856.

[116] A. Ashukha, A. Lyzhov, D. Molchanov, D. Vetrov, Pitfalls of in-domain uncertainty estimation and ensembling in deep learning, in: ICLR, 2020, pp. 1–30.

[117] P. McDermott, C. Wikle, Deep echo state networks with uncertainty quantification for spatio-temporal forecasting, Environmetrics 30 (2019) e2553 (paper no.).

[118] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, et al., Google vizier: A service for black-box optimization, in: Knowledge Discovery and Data Mining, 2017, pp. 1487–1496.

[119] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: NeurIPS, 2014, pp. 1–9.

[120] E. Tzeng, J. Hoffman, T. Darell, K. Saenko, Simultaneous deep transfer across domains and tasks, in: ICCV, 2015, pp. 4068–4076.

[121] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, D. Erhan, Domain separation networks,

in: NeurIPS, 2016, pp. 1–9.

[122] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1717–1724.

[123] M. Long, H. Zhu, J. Wang, M. Jordan, Deep transfer learning with joint adaptation networks, in: ICML, 2017, pp. 3470–3479.

[124] W. Cui, G. Zheng, Z. Shen, S. Jiang, W. Wang, Transfer learning for sequences via learning to collocate, in: ICLR, 2019, pp. 1487–1496.

[125] F. Zhuang, X. Cheng, P. Luo, S. Pan, Supervised representation learning: Transfer learning with deep autoencoders, in: IJCAI, 2015, pp. 4119–4125.

[126] F. Zhuang, X. Cheng, P. Luo, S. Pan, Q. He, Supervised representation learning with double encoding-layer autoencoder for transfer learning, ACM Transactions on Intelligent Systems and Technology 9 (2018) 1–17.

[127] E. Tzeng, J. Hoffman, K. Saenko, T. Darell, Adversarial discriminative domain adaptation, in: CVPR, 2017, pp. 1–10.

[128] Z. Cao, M. Long, J. Wang, M. Jordan, Partial transfer learning with selective adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2724–2732.

[129] B. Wang, M. Qiu, X. Wang, Y. Li, Y. Gon, et al., A minimax game for instance based selective transfer learning, in: KDD, 2019, pp. 34–43.

[130] M. Liu, Coupled generative adversarial networks, in: NeurIPS, 2016, pp. 1–9.

[131] Y. Stoyanova, YanasGH/RAFs, 2023. https://github.com/YanasGH/RAFs.

[132] C. D. Archive, Algorithm data sets for the bbob test suite, 2023. https://numbbo.github.io/data-archive/bbob/.

[133] S. Garcia, F. Herrera, An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[134] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks?, Journal of Machine Learning Research 17 (2016) 1–10.

[135] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

A. Employed Benchmarks

The functions in the bbob suite are divided into five groups:

1. Separable functions (Figure 4).
   - f_1: sphere;
   - f_2: ellipsoidal;
   - f_3: Rastrigin;
   - f_4: Büche-Rastrigin;
   - f_5: linear slope.
2. Functions with low or moderate conditioning (Figure 5).
   - f_6: attractive sector;
   - f_7: step ellipsoidal;
   - f_8: Rosenbrock;
   - f_8: Rosenbrock rotated.
3. Unimodal functions with high conditioning (Figure 6).
   - f_{10}: ellipsoidal;
   - f_{11}: discus;
   - f_{12}: bent cigar;

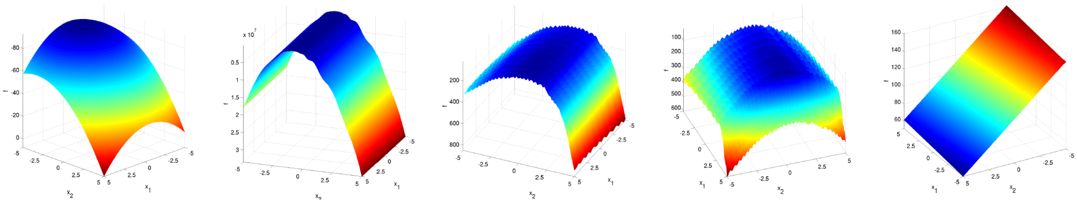

Figure 4: Separable functions. From left to right: sphere, ellipsoidal, Rastrigin, Büche-Rastrigin, maximized linear slope.

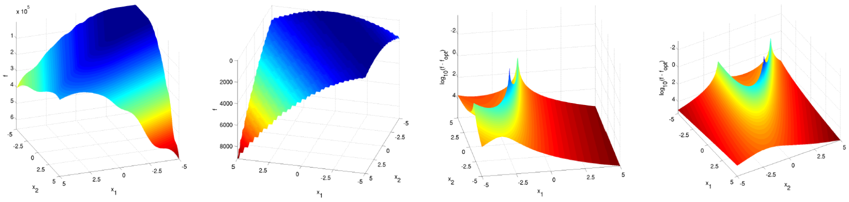

Figure 5: Functions with low or moderate conditioning. From left to right: attractive sector, step ellipsoidal, Rosenbrock, Rosenbrock rotated.

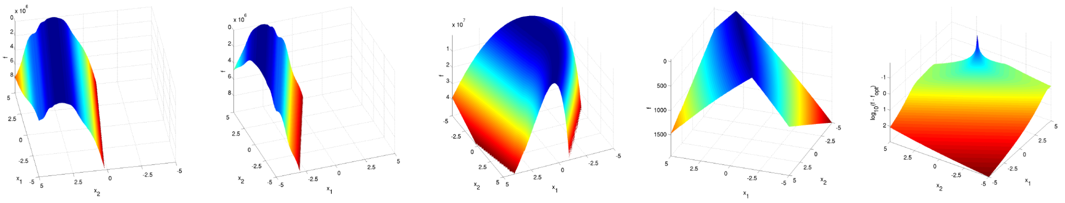

Figure 6: Unimodal functions with high conditioning. From left to right: ellipsoidal, discus, bent cigar, sharp ridge, different powers.

- f_{13}: sharp ridge;
- f_{14}: different powers.

4. Multi-modal functions with adequate global structure (Figure 7).
   - f_{15}: Rastrigin;
   - f_{16}: Weierstrass;
   - f_{17}: Schaffers F7 function;
   - f_{18}: Schaffers F7 function, moderately ill-conditioned;
   - f_{19}: composite Griewank-Rosenbrock function F8F2.

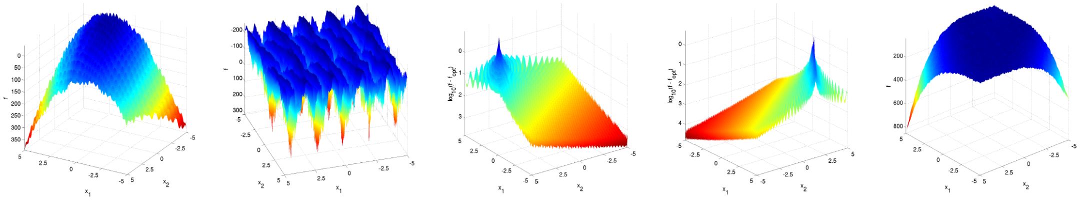

Figure 7: Multi-modal functions with adequate global structure. From left to right: Rastrigin, Weierstrass, Schaffers F7 function, moderately ill-conditioned Schaffers F7 function, composite Griewank-Rosenbrock function F8F2.

5. Multi-modal functions with weak global structure (Figure 8).
   - f_{20}: Schwefel;
   - f_{21}: Gallagher's Gaussian 101-me peaks;
   - f_{22}: Gallagher's Gaussian 21-hi peaks;
   - f_{23}: Katsuura;
   - f_{24}: Lunacek bi-Rastrigin.

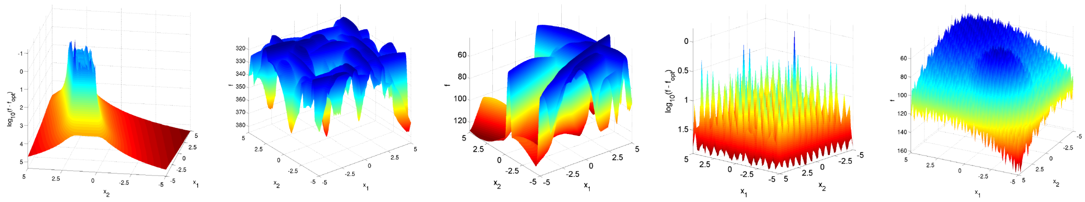

Figure 8: Multi-modal functions with weak global structure. From left to right: Schwefel, Gallagher's Gaussian 101-me peaks, Gallagher's Gaussian 21-hi peaks, Katsuura, Lunacek bi-Rastrigin.

B. Activation Functions Employed to Form an RAF Ensemble

- *Gauss error function*

$$\mathrm{erf}(x) = \begin{cases} \int_0^x \mathrm{e}^{-t^2} \mathrm{d}t & \text{if } x \geq 0, \\ -\int_0^{-x} \mathrm{e}^{-t^2} \mathrm{d}t & \text{if } x < 0. \end{cases} \tag{2}$$

- *Gaussian error linear unit*

$$\mathrm{gelu}(x) = \frac{x}{2}(1 + \mathrm{erf}(\frac{x}{\sqrt{2}})). \tag{3}$$

- *Scaled exponential linear unit*

$$\mathrm{selu}(x) = \begin{cases} cx & \text{if } x \geq 0, \\ c\alpha(\mathrm{e} - 1), \end{cases} \tag{4}$$

  where $c, \alpha > 0$. In the employed Tensorflow implementation, $c = 1.05070098, \alpha = 1.67326324$.
- *Softsign* activation function

$$\mathrm{softsign}(x) = \frac{x}{|x| + 1}. \tag{5}$$

- *Hyperbolic tangent*

$$\tanh(x) = \frac{\mathrm{e}^x - \mathrm{e}^{-x}}{\mathrm{e}^x + \mathrm{e}^{-x}}. \tag{6}$$