Comparing Non-Linear Regression Methods on Black-Box Optimization Benchmarks

Vojtěch Kopal[1] Martin Holeňa[2]

[1] Charles University in Prague, Faculty of Mathematics and Physics,
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
vojtech.kopal@gmail.com,
[2] Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, 182 07 Praha, Czech Republic
martin@cs.cas.cz

Abstract: The paper compares several non-linear regression methods on synthetic data sets generated using standard benchmarks for continuous black-box optimization. For that comparison, we have chosen regression methods that have been used as surrogate models in such optimization: radial basis function networks, Gaussian processes, and random forests. Because the purpose of black-box optimization is frequently some kind of design of experiments, and because a role similar to surrogate models is in the traditional design of experiments played by response surface models, we also include standard response surface models, i.e., polynomial regression. The methods are evaluated based on their mean-squared error and on the Kendall's rank correlation coefficient between the ordering of function values according to the model and according to the function used to generate the data.

1 Introduction

In this paper, we compare non-linear regression methods that could be used as surrogate models for optimization tasks. The methods are compared on synthetic data sets generated using standard benchmarks for continuous black-box optimization, for which we used implementations based on definitions from *Real-Parameter Black-Box Optimization Benchmarking 2009* [8].

A continuous black-box optimization is a task where we try to minimize a continuous objective function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ for which we do not have an analytical expression. Such problems arise, for example, if the values of the objective function are results of experimental measurements.

For that comparison, we have chosen regression methods that have been used as surrogate models in such optimization: *radial basis function networks* [3] [18], *Gaussian processes* [6] [11], and *random forests* [4].

We measure the accuracy of each methods based on mean square error and Kendall's rank coefficient and based on the results we suggest which methods work better as surrogate models. We are interested in properties of each method to be used as a surrogate model, though our experiments do not replace a direct evaluation in optimization or in evolutionary algorithms. This is a subject of two other papers included in this proceedings.

Other comparisons of non-linear models have been presented. A numerical comparison of neural networks and polynomial regression has been performed in [2] and in [16], in the latter one also classification and regression tree (CART) model has been compared. An evaluation of Gaussian processes with other non-linear methods has been done in [15] and in [10]. These studies compared accuracy of each model for prediction and have not paid attention to surrogate models for optimization. Example of such works can be found in [7], where they have compared quadratic polynomial regression with other methods based on prediction accuracy and mean-squared error, and in [13], where is polynomial regression compared with radial basis function networks based on accuracy and also on optimization results. In this paper, we compare the methods by means of mean-squared error and also Kendall's rank coefficient.

We briefly describe the theoretical background for each of these methods: how the corresponding models are being induced and how they are used to predict new values. For the synthetic data, we added an overview of how the functions look like in a 3-dimensional space (Figure 1).

The paper is organised as follows. In Section 2, we recall the theoretical fundamentals of the employed regression methods. In Section 3, we describe the setup of our experiments and summarise the results, before the paper concludes in Section 4.

2 Regression Methods in Data Mining

With a continuously increasing amount of gathered data, data mining techniques allow us to search for patterns in the data sets and model the underlying reality. Various models have been introduced in the past, starting from a linear regression to complex nonlinear methods such as neural networks, or Gaussian processes. These models are used to approximate a function that describes the relationship between target and input values.

We now introduce the methods compared in this paper. Each of these methods has its strengths and weaknesses

	Parameters	Hyper-parameters	Strengths	Weaknesses	Complexity
Polynomial regression	order of the polynom	×	fast and simple	too simple	$\Theta(M^2N)$
Gaussian processes	covariance function, mean function (usually constantly zero)	depends on cov. fun. length-scale (l) noise-level (SN)	robust, generalising well	time complexity, black box	$\Theta(N^3)$
Random forests	# of trees (NT), min. data in leaves (ML), # of randomly selected variables	×	interpretability, allows parallel computations	slower to compute predictions	$\Theta(MK\tilde{N}\log^2\tilde{N})$ [12]
Radial basis functions network	spread constant (CS) maximum of neurons (MAX) error goal (EG)	×	robust	black box	polynomial time [17]

Table 1: Summary of regression methods. *N = # of samples, $\tilde{N} = 0.632N$, M = # of variables, K = # of variables randomly drawn at each node (in random forests)*

which we point out in Table 2 and later we will discuss them in the context of results of our experiments.

We assume to have a pair $((X),Y)$, where $\mathbf{X}$ is p-dimensional data set with n points, i.e. $\mathbf{X}$ is a matrix $p \times n$, or it is a vector $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$, where $\mathbf{x_i}$ is a column vector of size p, i.e. $\mathbf{x_i} = (x_{i1}, x_{i2}, ..., x_{ip})$, and $Y = (Y_1, Y_2, ..., Y_n)$ is a vector of size n of target values to corresponding rows in matrix $\mathbf{X}$. We use $||\mathbf{x}||$ as the Euclidean norm of vector $\mathbf{x}$. In the paper, we use following notation:

- X, Y, β are vectors with elements X_i, Y_i, β_i, respectively, also $\beta_{j,k}$ is a scalar denoting a parameter in polynomial regression for interaction $x_j x_k$,
- f is a function and $f(x)$ is an output of the functions corresponding to input x, for multivariate function f, we have either matrix notation $Y = f(\mathbf{X})$, or vector notation $Y_i = f(\mathbf{x}_i)$,
- $\bar{f}(\mathbf{X})$ is an average output over $f(\mathbf{x}_i), \forall i \in \{1, ..., N\}$.

2.1 Polynomial Regression

The most simple form of *polynomial regression* (PR) is linear regression in which the model is described by $p+1$ parameters $\beta_0, \beta_1, ..., \beta_p$,

$$f(X_i) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j$$

which can be computed by [9]

$$\beta = (\beta_0, \dots, \beta_p) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

Polynomial regression is still part of the linear regression family, because the dependence on the model parameters is linear. However, we consider also higher powers of input variables. For example, in the quadratic case we add both x_i^2 form for $i \in \{1, ..., p\}$, and also as an interaction $x_i x_j$ for $i, j \in \{1, ..., p\}, i \neq j$. Consequently we have $(p^2 + p)/2$ new variables.

For our experiments, we will restrict attention to quadratic regression,

$$f(X_i) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j + \sum_{j=1}^{p} x_j^2 \beta_{p+j} + \sum_{j=1}^{p} \sum_{\substack{k=1 \\ k<j}}^{p} x_j x_k \beta_{i,k}$$

This is also the standard restriction in response surface modeling [14].

2.2 Random Forests

Random Forests (RF) is a model proposed by Breiman [5], and it is based on ensembles of decision trees. Due to our interest in surrogate models for continuous black-box optimization, we are interested in ensembles of regression trees.

A regression tree is a function defined by means of a binary tree with inner nodes representing predicates, and edges from a node to its children representing whether the predicate is or is not fulfilled. The leaf nodes give the predicted target value. The tree is built recursively starting with a root node and searching for an optimal binary predicate over the input variables. Regression trees can be applied to data sets with both categorical/discrete variables, and real-valued variables. Since we focus on surrogate models for continuous black-box optimization, we only consider real-valued predicates. For a real-valued variable, the data set is split into two parts through minimizing following formula

$$\sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2$$

where R_1, R_2 are the two linearly bounded regions with axes-perpendicular borders into which the data set is split using a j-th variable x_j and its splitting point s, and c_1, c_2

are the averages of function values of points belonging to R_1, R_2, respectively. After finding the optimal splitting point we recursively apply this process to both regions R_1 and R_2, and for each of them, only the data points in the region are considered. This process continues until a stopping criterion is met. This can be either the *minimum number of data points in leaves* or inner nodes, or the depth of the tree.

If the regression tree finally splits the input space into the regions $R_1, \ldots, R_m$, we can compute the prediction for a new data point using the following formula:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

where c_m is an average target value of data points in region R_m.

An ensemble of regression trees averages the predictions when presented with a new data point.

There are several options how to induce a number of trees over the same data set that will lead to low correlation. In traditional bagging, independent subsets of the original data used for individual trees are obtained by sampling from the data set uniformly and with replacement. In addition, random subsets of input variables can be used. In Matlab implementation of random forests, a square root of number of input variables are selected by default, which is also a setting we have used for our experiments.

The model parameters are *number of trees* (NT) which are added to the ensemble and the minimum number of data in leaves (ML).

2.3 Gaussian Processes

A *Gaussian process* (GP) is a random process such that its restriction to any finite number of points has a Gaussian probability distribution. A Gaussian process $\mathbb{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$ is defined by its mean function $\mu(\mathbf{x})$ and a covariance function $\kappa(\mathbf{x}, \mathbf{x}')$.

$$f(\mathbf{x}) \sim \mathbb{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')) \tag{2}$$

These functions determine the mean and covariance of the process because

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= \mu(\mathbf{x}), \\ Cov[f(\mathbf{x}), f(\mathbf{x}')] &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] \\ &= \kappa(\mathbf{x}, \mathbf{x}') \end{aligned} \tag{3}$$

The important part of modelling functions with Gaussian processes is choosing the covariance function. An important feature of covariance functions is that they can be combined together using addition and multiplication, i.e. for κ, κ' covariance functions, $\kappa \times \kappa'$ and $\kappa + \kappa'$ are again covariance functions. Frequently used covariance functions are: linear, periodic, squared-exponential, and rational quadratic.

- **Linear**:
$$\kappa_{lin}(\mathbf{x}, \mathbf{x}') = \mathbf{x}\mathbf{x}'$$
- **Periodic**:
$$\kappa_{per}(r) = exp(-\frac{2}{l^2} sin^2(\pi \frac{r}{p}))$$
- **Squared-exponential**:
$$\kappa_{SE}(r) = exp(-\frac{r^2}{2l^2})$$
- **Rational Quadratic**:
$$\kappa_{RQ}(r) = (1 + \frac{r^2}{2\alpha l^2})^{-\alpha}$$

where $r = |\mathbf{x} - \mathbf{x}'|$ and c, l, p, α are parameters of the covariance function (because the covariance function itself is a parameter of the Gaussian process, they are called hyper-parameters of the process). l is a length-scale, p defines period, and α changes the smoothness of rational quadratic function. An additional parameter in the model is the *noise level* (SN) which is an additive Gaussian noise in the model.

When working with multivariate data sets, the covariance functions which have the *length-scale* as a parameter, can either apply the same length-scale l to all dimensions, or i-th dimension has its length-scale l_i. In the first case, the covariance functions have isotropic distance measure, the latter case uses *automatic relevance determination (ARD)*.

2.4 Radial Basis Network Functions

Radial basis network functions (RBF) is a feed-forward neural network with one hidden layer in which the nodes have radial transfer function ρ. The output of the network is given by

$$\varphi(\mathbf{x}) = \sum_{i=1}^{N} a_i \rho(||\mathbf{x} - \mathbf{c_i}||) \tag{4}$$

or its normalized version:

$$\varphi(\mathbf{x}) = \frac{\sum_{i=1}^{N} a_i \rho(||\mathbf{x} - \mathbf{c_i}||)}{\sum_{i=1}^{N} \rho(||\mathbf{x} - \mathbf{c_i}||)} \tag{5}$$

where $\rho(||\mathbf{x} - \mathbf{c_i}||)$ is usually in form of *gaussian*:

$$\rho(||\mathbf{x} - \mathbf{c_i}||) = \exp\left(\frac{||\mathbf{x} - \mathbf{c_i}||^2}{2\sigma_i^2}\right),$$

$\mathbf{c_i}$ is a center vector of the respective neuron, a_i is a weight of the neuron, and $||\mathbf{x} - \mathbf{c_i}||$ is a norm, typically the Euclidean norm. The model parameters are the *spread constant* σ_i^2 (SC), the *maximum of neurons* (MAX) that can be added to network during iterative learning process, and the *error goal* (EG) which is a mean-squared error on training set. The maximum neurons or the error goal are stopping criteria for the network induction.

2.5 Model Selection and Evaluation

The parameters for regression models were selected by 10-fold cross validation based on the *mean-squared error (MSE)*

$$\overline{err} = MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{f}(\mathbf{X}))^2$$

The cross validation is suited for limited data samples, but it is also justified method for synthetic data.

3 Experiments with Synthetic Data

As we are interested primarily in the suitability of the considered regression methods for surrogate models in black-box optimization, we compared them on synthetic data generated using standard benchmarks for continuous black-box optimization [8].

All performed experiments were implemented in Matlab. For each function, we have sampled 5000 p-dimensional data points where $p \in \{5, 10, 20, 40\}$ and used it for a 10-fold cross-validation to compare the considered models. The result of cross validation is MSE for training set, MSE for testing set and the Kendall's rank correlation coefficient. The significance of the difference between results obtained for two models m, m' was tested using independent sample t-test

$$t = \frac{\overline{res}_m - \overline{res}_{m'}}{\sqrt{\frac{1}{k}(\sigma_m^2 + \sigma_{m'}^2)}} \tag{6}$$

which we compare for a significance level $\alpha \in (0,1)$ against the $(1-\frac{\alpha}{2})$-quantile of the Student distribution with 2(k-1) degrees of freedom, where k is the number of cross-validation folds, degrees of freedom, and $\overline{res}_m, \sigma_m$ are computed as follows:

$$\overline{res}_m = \frac{1}{k}\sum_{i=1}^{k} res_{m,i}$$

$$\sigma_m = \sqrt{\frac{1}{(k-1)}\sum_{i=1}^{k}(res_i - \overline{res})^2}$$

For a comparison of two models, it would have been better to use paired t-test, which provide better estimates, but since we have decided to use unpaired t-test at the beginning of our experiment, we haven't had necessary subresults to perform it.

We have used MSE together with *Kendall's rank correlation coefficient* [1] between the ordering of function values according to the model $(y_1, \ldots, y_n)$ and according to the function used to generate the data $(t_1, \ldots, t_n)$

$$\tau_m = \frac{(\text{\# of concordant pairs}) - (\text{\# of discordant pairs})}{\frac{1}{2}n(n-1)} \tag{7}$$

where for (t_j, y_j) and (t_k, y_k) different pairs of target value t and predicted value y, (t_j, y_j) and (t_k, y_k) are concordant if $t_j < t_k$ and $y_j < y_k$, or $t_j > t_k$ and $y_j > y_k$, and discordant otherwise.

3.1 Selection of Model Parameters

For each dataset, we have searched for optimal model parameters (in the case of a Gaussian process, these are its hyper-parameters) minimizing the MSE. With regression trees, we have considered different settings for the number of trees and minimum number of data points in leaves. With Gaussian processes, we have tried rational quadratic and squared exponential in their isomorphic form, and also the ARD version of squared exponential. With radial basis function networks, as a radial function, we have considered different settings of the parameters: spread constant, MSE goal, maximum of neurons. As to polynomial regression, we have used quadratic regression. See Table 3 for overview of selected parameters for each model.

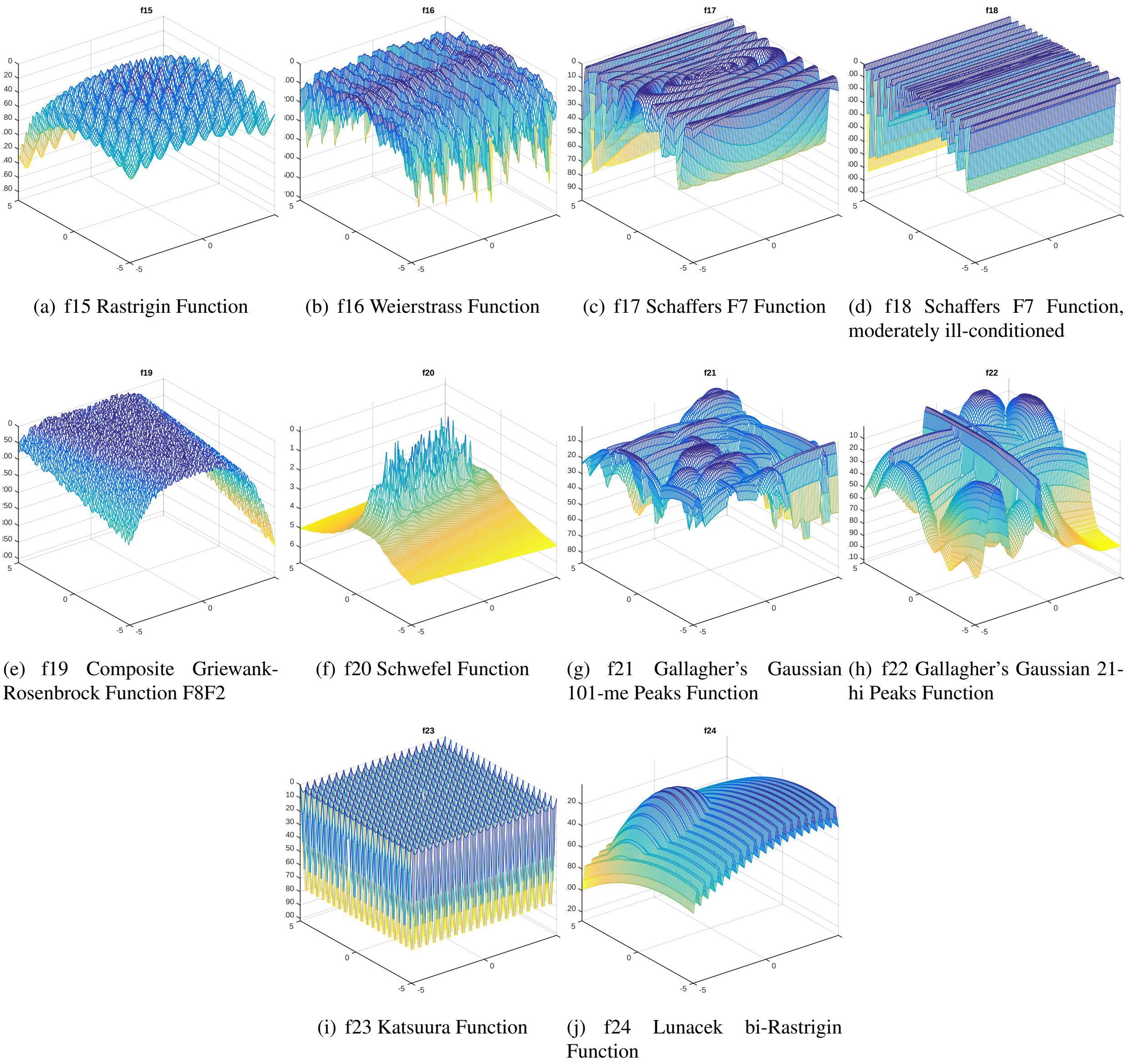
3.2 Results

We will now present the results of our experiments. First, we have included a detailed Table 3 with measured values of the MSE and the Kendall's coefficient for each dataset and each model. We can see how optimal combinations of values of parameters for each model change with higher dimensions. Random forests have lower number of trees (NT) and higher minimum number of data in leaves (ML). The comparison of the performance of each method across different dimensions of the data sets follows.

Table 4 shows the results of our experiments where we have compared four different models across 40 different data sets. For each model, we entered the number of times the model was better than the other model and we also added how many times the result was significantly better on the significance level 0.05.

A summary of the results can be seen in Table 2 and additional comments on the results follow. With 10 dimensions, the radial basis functions started performing better, although not significantly. With 20 dimensions, there are even less methods that outperforms polynomial regression significantly according to MSE and random forests were the weakest model from the triple of models RBF, GP and RT. With 40 dimensions, there is a surprising result since MSE values are much lower comparing to lower dimensions and we would expect the MSE to be growing with higher dimensions. This may be an artifact of the function definitions which suppress higher dimensions and that may lower the MSE values.

In summary, when comparing the MSE over all dimensions, the Gaussian processes were the best model for our data followed by radial basis functions and random forests before polynomial regression at the last position. With Kendall's coefficient, the results are not that clear. Even though the Gaussian processes have the most wins, they do



(a) f15 Rastrigin Function

(b) f16 Weierstrass Function

(c) f17 Schaffers F7 Function

(d) f18 Schaffers F7 Function, moderately ill-conditioned

(e) f19 Composite Griewank-Rosenbrock Function F8F2

(f) f20 Schwefel Function

(g) f21 Gallagher's Gaussian 101-me Peaks Function

(h) f22 Gallagher's Gaussian 21-hi Peaks Function

(i) f23 Katsuura Function

(j) f24 Lunacek bi-Rastrigin Function

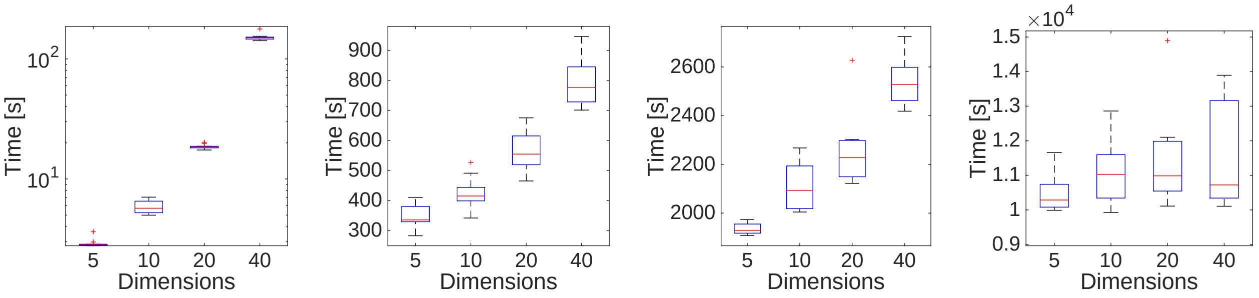
Figure 1: Benchmark functions for continuous black-box optimization. The graphs has been created in accordance with [8]. Since we has been focused on multimodal functions, we chose functions 15 to 24.

Dimension	Mean squared error	Kendall's rank correlation coefficient
5	GP outperformed other models in most of the cases, all methods performs better most of the time comparing to polynomial regression	less conclusive results, the best performing model were RF
10	GP outperformed other models	less conclusive results, the best performing model were RF
20	GP outperformed other models	RBF was the weakest model outperformed even by PR
40	GP still outperformed others, but with smaller difference	GP outperformed others

Table 2: Results summary. For each dimension, we briefly describe the outcome.

Dataset	Random forests				Gaussian Processes				Polynomial		Radial basis functions				
	NT	ML	MSE	Kendall	CF	SN	MSE	Kendall	MSE	Kendall	SC	EG	MAX	MSE	Kendall
f15-05d	1000	1	**2.1±0.2e2**	**7.1±0.1e-1**	SEiso	1	2.6±0.1e2	6.6±0.2e-1	*3.1±0.2e2*	*6.4±0.2e-1*	8e1	1	500	2.5±0.1e2	6.7±0.2e-1
f15-10d	1600	1	*6.9±0.5e2*	*6.4±0.2e-1*	SEiso	1	5.8±0.5e2	**6.7±0.2e-1**	6.2±0.3e2	6.5±0.2e-1	5e1	1	250	**5.7±0.4e2**	6.6±0.1e-1
f15-20d	1200	1	*1.9±0.2e3*	*5.7±0.2e-1*	RQiso	5e-1	1.4±0.1e3	6.3±0.1e-1	1.31±0.07e3	**6.4±0.2e-1**	1e2	1	150	**1.28±0.06e3**	6.4±0.2e-1
f15-40d	800	2	*5.1±0.2e3*	*4.9±0.2e-1*	RQiso	5e-1	3±0.2e3	6.1±0.1e-1	3±0.2e3	6.1±0.2e-1	5e1	1	75	**2.7±0.2e3**	**6.3±0.1e-1**
f16-05d	1600	1	**1.1±0.1e3**	**5.3±0.1e-1**	SEiso	1	1.8±0.1e3	3.4±0.3e-1	*1.9±0.1e3*	*2.7±0.3e-1*	7.5e1	1	800	1.7±0.2e3	3.5±0.4e-1
f16-10d	800	1	**7.1±0.5e2**	**4.3±0.3e-1**	SEiso	1	*9.4±0.08e2*	*2±1e-1*	8.6±0.6e2	2.7±0.2e-1	1.8e1	1	40	9±1e2	2.5±0.3e-1
f16-20d	800	2	**4.1±0.4e2**	**3.6±0.2e-1**	SEiso	1e-2	*7.8±0.2e3*	*-4±4e-2*	4.6±0.3e2	2.3±0.2e-1	3.2e1	1	60	5±0.3e2	1.3±0.4e-1
f16-40d	400	5	**2.3±0.2e2**	**2.9±0.4e-1**	RQiso	5e-1	2.5±0.2e2	8±3e-2	*2.7±0.2e2*	1.7±0.2e-1	7.5	1	50	2.5±0.2e2	*7±2e-2*
f17-05d	1800	1	**2.3±0.3e1**	**4.6±0.2e-1**	SEiso	1.3	3.2±0.3e1	2.2±0.3e-1	3.3±0.2e1	*2.1±0.2e-1*	4.8	1	60	*3.3±0.2e1*	2.1±0.2e-1
f17-10d	800	1	**1.35±0.07e1**	**3.2±0.3e-1**	SEiso	1	*1.5±0.1e1*	*2.2±0.2e-1*	1.48±0.09e1	2.2±0.4e-1	4e1	1	50	1.4±0.1e1	2.4±0.3e-1
f17-20d	800	1	**6.8±0.4**	**2.6±0.3e-1**	SEiso	1e-2	*6.5±0.2e1*	*-1.3±0.3e-1*	6.9±0.5	2.2±0.3e-1	6.4e1	1	250	7.2±0.4	2±0.3e-1
f17-40d	400	1	3.4±0.2	2.2±0.3e-1	RQiso	5e-1	**3.2±0.2**	2±0.4e-1	*4.1±0.3*	*1.3±0.2e-1*	8e1	1e-3	40	3.3±0.2	**2.3±0.3e-1**
f18-05d	1200	1	**5.2±0.3e2**	**4.9±0.3e-1**	SEiso	2	8±1e2	2.4±0.2e-1	8.5±0.7e2	*2.3±0.3e-1*	5e1	1	500	*9.2±0.08e2*	2.3±0.2e-1
f18-10d	800	2	**2.7±0.2e2**	**3±0.2e-1**	SEiso	1	2.9±0.2e2	2.3±0.3e-1	*2.9±0.3e2*	*2.3±0.2e-1*	6e1	1	50	2.9±0.2e2	2.3±0.3e-1
f18-20d	1200	1	**1.26±0.06e2**	**2.4±0.4e-1**	SEiso	1e-2	*1.06±0.03e3*	*-1.3±0.3e-1*	1.41±0.05e2	2.2±0.2e-1	1e-3	1e-3	20	1.5±0.1e2	0±0
f18-40d	400	2	**6.4±0.5e1**	**2.1±0.2e-1**	RQiso	5e-1	6.5±0.4e1	1.9±0.2e-1	*7.6±0.6e1*	*1.5±0.3e-1*	8e1	1e-3	40	6.4±0.5e1	2±0.2e-1
f19-05d	800	50	5±0.3e2	5.2±0.2e-1	RQiso	5e-1	**4.4±0.3e2**	**5.4±0.2e-1**	4.8±0.3e2	5±0.2e-1	1e-2	1	500	*1.08±0.08e3*	*0±0*
f19-10d	2000	1	4.5±0.3e2	3.8±0.3e-1	RQiso	5e-1	**4.4±0.3e2**	**3.9±0.3e-1**	4.4±0.3e2	3.8±0.3e-1	1e-2	1	500	*6.8±0.4e2*	*0±0*
f19-20d	800	50	4.6±0.2e2	2.3±0.2e-1	RQiso	5e-1	**4.3±0.3e2**	**2.7±0.2e-1**	4.4±0.3e2	2.7±0.3e-1	1e-1	1	1000	*5.9±0.3e2*	*0±0*
f19-40d	800	50	4.3±0.3e2	1.6±0.3e-1	RQiso	5e-1	**4.2±0.1e2**	**1.8±0.2e-1**	*5.1±0.3e2*	1.3±0.4e-1	5	1	50	4.5±0.2e2	*-1.3±0.3e-1*
f20-05d	2000	1	1.2±0.8e-2	9.23±0.04e-1	SEiso	1	**1.1±0.8e-2**	**9.35±0.04e-1**	*7±3e-2*	*8.5±0.06e-1*	4.5	1e-3	1000	1.1±0.6e-2	9.13±0.06e-1
f20-10d	1800	1	8±1e-3	*8.3±0.1e-1*	SEiso	1e-1	**3.5±0.7e-3**	**9.18±0.03e-1**	*1.2±0.2e-2*	8.98±0.05e-1	8.6	1e-4	1150	3.8±0.5e-3	8.92±0.07e-1
f20-20d	1800	1	*8.6±0.6e-3*	*7±0.1e-1*	RQiso	5e-1	**1.5±0.2e-3**	9.01±0.04e-1	2.4±0.3e-3	**9.1±0.05e-1**	1.5e1	1e-4	800	1.8±0.2e-3	8.84±0.06e-1
f20-40d	800	5	*6.9±0.5e-3*	*5.99±0.07e-1*	RQiso	5e-1	**7.7±0.6e-4**	**9.03±0.05e-1**	8.3±0.7e-4	9.03±0.06e-1	8e1	1e-4	700	8.1±0.6e-4	9.02±0.06e-1
f21-05d	1600	1	1.4±0.2e2	5.7±0.3e-1	RQiso	5e-1	**6.7±0.6e1**	**6.9±0.2e-1**	*2.5±0.3e2*	*1.9±0.5e-1*	2	1	1800	9.6±0.09e1	6.1±0.2e-1
f21-10d	800	5	3±0.6e1	4.1±0.3e-1	SEiso	1	**2.4±0.4e1**	**5.1±0.2e-1**	*3.1±0.6e1*	*3.9±0.2e-1*	8	1	600	2.5±0.3e1	4.9±0.2e-1
f21-20d	800	10	3±0.4	3.5±0.3e-1	SEiso	1e-2	*7.187±0.009e3*	*-2.1±0.1e-1*	2.6±0.4	**4.6±0.3e-1**	8	1	800	**2.4±0.4**	4.4±0.3e-1
f21-40d	200	20	*2.5±0.9e-1*	*2.9±0.3e-1*	RQiso	5e-1	**1.8±0.7e-1**	**5.3±0.3e-1**	2±0.6e-1	4.8±0.3e-1	1.5e1	1e-3	400	2±0.7e-1	4.6±0.3e-1
f22-05d	1800	1	5.1±0.7e1	7.1±0.1e-1	SEiso	1	**3.5±0.5e1**	**7.4±0.2e-1**	*1.3±0.2e2*	*4.1±0.3e-1*	3	1	700	3.8±0.5e1	7±0.2e-1
f22-10d	800	2	9±0.3	5.3±0.3e-1	SEiso	1e-1	**8±3**	**5.9±0.1e-1**	*1.1±0.3e1*	*5.1±0.3e-1*	1e1	1	500	9±3	5.2±0.1e-1
f22-20d	800	10	7±1e-1	4.5±0.1e-1	SEiso	1e-2	*7.385±0.005e3*	*-2.1±0.2e-1*	6±1e-1	**6±0.2e-1**	5.8	1e-2	1500	**6±1e-1**	5.9±0.1e-1
f22-40d	200	20	*5±1e-2*	*3.5±0.2e-1*	RQiso	5e-1	**3±0.8e-2**	**6±0.2e-1**	3.6±0.8e-2	5.5±0.2e-1	2e1	1e-3	400	3.7±0.9e-2	5.2±0.1e-1
f23-05d	1200	1	**6.7±0.3e1**	**3.7±0.2e-1**	SEiso	5e-1	*8.5±0.5e1*	0±0	8.4±0.4e1	*-1±3e-2*	1e-3	1e-3	200	8.4±0.5e1	0±0
f23-10d	1000	1	**4.1±0.2e1**	**2±0.4e-1**	SEiso	1.2	*4.5±0.2e1*	1±40e-2	4.4±0.1e1	1±30e-2	1.2	1e-3	100	4.4±0.3e1	*0±300*
f23-20d	200	2	**2.7±0.2e1**	**1±0.3e-1**	SEiso	1e-2	*2.63±0.07e2*	*-1±30e-2*	2.9±0.2e1	2±4e-2	1e-3	1e-3	5	2.7±0.2e1	0±0
f23-40d	800	10	1.49±0.09e1	**6±2e-2**	RQiso	5e-1	1.54±0.07e1	2±2e-2	*1.8±0.1e1*	3±3e-2	5	1	50	**1.5±0.1e1**	*1±30e-2*
f24-05d	1200	1	*4.5±0.4e2*	*7.2±0.1e-1*	SEiso	1.2	**2.7±0.2e2**	**7.67±0.1e-1**	4.1±0.4e2	7.2±0.1e-1	5.5e1	1e-3	400	2.8±0.2e2	7.6±0.1e-1
f24-10d	800	2	*2.3±0.1e3*	*6.4±0.1e-1*	SEiso	7.5e-1	9±5e2	7.8±0.4e-1	9±0.05e2	7.7±0.1e-1	1.2e1	1e-3	300	**7±0.5e2**	**7.94±0.09e-1**
f24-20d	800	2	*8.9±0.7e3*	*5.9±0.2e-1*	RQiso	5e-1	1.8±0.1e3	8±0.1e-1	1.8±0.1e3	8±0.1e-1	8e1	1	300	**1.8±0.2e3**	**8±0.1e-1**
f24-40d	400	5	*2.8±0.2e4*	*5.2±0.3e-1*	RQiso	1	4.6±0.3e3	7.9±0.1e-1	**4.5±0.4e3**	**7.9±0.1e-1**	8e1	1e-3	400	5.8±0.4e3	7.6±0.1e-1

Table 3: The values of MSE and Kendall's rank correlation coefficient for each dataset and that combination of model parameters used in each considered method that lead to the minimal test-data MSE for the performed crossvalidation of the respective dataset. The best results are shown in bold, the worst results in italics. *Random forests parameters: NT = number of trees, ML = minimum of data in leaves. Gaussian Processes parameters: CF = covariance function, SN = noise level (hyperparameter, additional hyperparameters for GP covariance functions were omitted). Radial basis functions parameters: SC = spread constant, EG = error goal, MAX = maximum neurons.*



(a) Polynomial regression (b) Random forests (c) Radial basis function networks (d) Gaussian processes

Figure 2: An overview how long does it take to compute 10-fold cross validation for each regression model and each of the considered dimensions of data. (a) Polynomial regression (quadratic) (b) Random forests (number of trees: 800, minimum number of data in leaves: 1) (c) Radial basis function networks (spread constant: 5.5, mean square error goal: 1, maximal number of neurons: 500) (d) Gaussian processes (covariance functions: rational quadratic (isomorphic), noise level: $sn = 0.5$, length-scale: $ell = log(1.1)$, signal variance $sf2 = log(1)$, shape parameter: $\alpha = log(1.1)$)

Dimension	Method	Mean squared error				Kendall's rank correlation coefficient			
		GP	RBF	RF	Polynom	GP	RBF	RF	Polynom
5	GP	×	9 (3)	5 (3)	10 (6)	×	8 (5)	4 (4)	10 (6)
	RBF	1 (0)	×	5 (3)	8 (6)	2 (0)	×	2 (2)	8 (6)
	RF	5 (5)	5 (5)	×	8 (8)	6 (5)	8 (6)	×	9 (8)
	Poly	0 (0)	2 (0)	2 (0)	×	0 (0)	2 (0)	1 (0)	×
10	GP	×	6 (2)	5 (4)	8 (4)	×	7 (3)	5 (5)	9 (6)
	RBF	4 (0)	×	6 (4)	9 (4)	3 (0)	×	5 (4)	10 (3)
	RF	5 (2)	4 (3)	×	8 (5)	5 (4)	5 (4)	×	7 (7)
	Poly	2 (0)	1 (0)	2 (2)	×	1 (0)	0 (0)	3 (2)	×
20	GP	×	6 (2)	6 (5)	6 (3)	×	8 (5)	5 (5)	5 (2)
	RBF	4 (1)	×	6 (4)	7 (1)	2 (1)	×	5 (5)	2 (0)
	RF	4 (3)	4 (3)	×	5 (3)	5 (3)	5 (4)	×	5 (4)
	Poly	4 (1)	3 (2)	5 (3)	×	5 (2)	8 (3)	5 (5)	×
40	GP	×	7 (1)	7 (5)	8 (6)	×	5 (3)	5 (5)	6 (4)
	RBF	3 (1)	×	7 (3)	8 (6)	5 (3)	×	6 (5)	4 (3)
	RF	3 (0)	3 (1)	×	5 (5)	5 (4)	4 (2)	×	5 (5)
	Poly	2 (0)	2 (1)	5 (3)	×	4 (1)	6 (2)	5 (5)	×
Summary	GP	×	28 (8)	23 (17)	32 (19)	×	28 (16)	19 (19)	30 (17)
	RBF	12 (2)	×	24 (14)	32 (17)	12 (4)	×	18 (16)	24 (12)
	RF	17 (10)	16 (12)	×	26 (21)	21 (16)	22 (16)	×	26 (24)
	Poly	8 (1)	8 (3)	14 (8)	×	10 (3)	16 (5)	14 (12)	×

Table 4: Results of experiments comparing 4 different models across 40 different data sets. For each model (in a row), we entered the number of times the model was better than the other model (in a column) and we also added how many times the result was significantly better on the significance level 0.05 (in the brackets).

not have the most significant wins. Based on the significant wins, the best performing model were random forests.

With higher dimensions, when comparing the models based on the MSE, we may notice that the results for Gaussian processes and random forests are less significant. Which is also the case with Kendall's coefficient, where the polynomial regression gets more wins with higher dimension.

Now we have a look at how long does it take to evaluate 10-fold cross validation for selected parameters settings for each model (see Figure 2). With higher dimensions, each method takes more time to evaluate. All the computations were performed on PC (x86-64) Intel Core i7 920 (4x 2.66 GHz + HyperThreading), 6 GB RAM.

4 Discussion and Conclusion

The figures and tables presented in Results compared four different regression methods over 40 synthetic data sets (10 functions $\times$ 4 different dimensions) generated using standard benchmark functions for continuous black-box optimization. We have shown how the performance of these methods changes with increasing dimensionality and how the time to cross-validate the models grows. We have compared the methods based on the MSE and on Kendall's coefficient. We will now comment on each of them.

Gaussian process is probably the most complex method. With its time complexity $O(N^3)$ it takes the longest time to compute, some of the cross-validations, i.e. 10 constructions of the model, took up to 24 hours. This model was better then the others according to both MSE and Kendall's coefficient comparison.

Random forests ended up with poorer results for 40 dimensional data and overall they were slightly behind Gaussian Processes based on the MSE. According to Kendall's coefficient results, they were comparable with Gaussian processes and, according to the number of significant wins, they even outperformed GP. With some data sets (f19-10d, f20-05d), we have learnt 2000 trees out of 4500 samples. In these cases, we could have compare the results with nearest neighbor method.

Radial basis functions network has the clearly poorer results compared to Gaussian processes and random forests according to both the MSE and the Kendall's coefficient.

Even though the *polynomial regression* was included due to its importance as traditional response surface model, the method was not always worse then all other methods. For the dimensions 20 and 40, their MSE was comparable to that of random forest. Also with higher dimensions, the results based on the Kendall's coefficient are comparable to both GP and RBF and it even outperformed RBF.

In this paper, we have compared a selection of nonlinear methods on synthetic data sets based on their mean-squared error and on the Kendall's rank correlation coefficient. We have chosen regression methods that have been used as surrogate models in such optimization: radial basis function networks, Gaussian processes, random forests, and polynomial regression. A better accuracy of the models suggests better applicability of the models as a surrogate model for optimization. From the results we have learnt that Gaussian processes had better results in most cases, thus, would be better surrogate model compared to the others, although random forests were only slightly behind.

Acknowledgements

This research was partially supported by SVV project number 260 224.

References

[1] Springer Encyclopedia of Mathematics

[2] Alessandri, A., Cassettari, L.,Mosca. R.: Nonparametric nonlinear regression using polynomial and neural approximators: a numerical comparison. Computational Management Science **6(1)** (2009), 5–24

[3] Bajer, L., Holeňa, M.: Surrogate model for continuous and discrete genetic optimization based on rbf networks. In: Colin Fyfe, Peter Tino, Darryl Charles, Cesar Garcia-Osorio, and Hujun Yin, (eds), Intelligent Data Engineering and Automated Learning – IDEAL 2010, volume 6283 of Lecture Notes in Computer Science, 251–258, Springer Berlin Heidelberg, 2010

[4] Bajer, L., Pitra, Z., Holeňa, M.: Benchmarking gaussian processes and random forests surrogate models on the bbob noiseless testbed. In: GECCO 2015, 2015

[5] Breiman, L.: Random forests. Mach. Learn. **45(1)** (October 2001), 5–32

[6] Buche, D., Schraudolph, N. N., Koumoutsakos, P.: Accelerating evolutionary algorithms with gaussian process fitness function models. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **35(2)** (May 2005), 183–194

[7] Gano, S. E., Kim, H., Brown, D. E.: Comparison of three surrogate modeling techniques: datascape, kriging and second order regression. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2006

[8] Hansen, N., Finck, S., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions. Research Report RR-6829, 2009

[9] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001

[10] Hultquist, C., Chen, G., Zhao, K.: A comparison of gaussian process regression, random forests and support vector regression for burn severity assessment in diseased forests. Remote Sensing Letters **5(8)** (2014), 723–732

[11] Kleijnen, J. P. C., van Beers, W., van Nieuwenhuyse, I.: Constrained optimization in expensive simulation: novel approach. European Journal of Operational Research **202(1)** (2010), 164 – 174

[12] Louppe, G.: Understanding random forests: from theory to practice. PhD thesis, 2014

[13] Luo, J., Lu, W.: Comparison of surrogate models with different methods in groundwater remediation process. Journal of Earth System Science **123(7)** (2014), 1579–1589

[14] Myers, R. H., Montgomery, D. C., Anderson-Cook, C. M.: Response surface methodology: proces and product optimization using designed experiments, 2009

[15] Rasmussen, C. E.: Evaluation of gaussian processes and other methods for non-linear regression. Technical Report, 1996

[16] Razi, M. A., Athappilly, K.: A comparative predictive analysis of neural networks (nns), nonlinear regression and classification and regression tree (cart) models. Expert Systems with Applications **29(1)** (2005), 65–74

[17] Roy, A., Govil, S. Miranda, R.: A neural-network learning theory and a polynomial time rbf algorithm. Neural Networks, IEEE Transactions on **8(6)** (Nov. 1997), 1301–1313

[18] Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J., Lum, K. Y.: Combining global and local surrogate models to accelerate evolutionary optimization. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **37(1)** (Jan. 2007), 66–76