CIplus
Band 6/2016

From Real World Data to Test Functions

Andreas Fischbach, Martin Zaefferer, Jörg Stork, Martina Friese, Thomas Bartz-Beielstein

Technology
Arts Sciences
TH Köln

From Real World Data to Test Functions

Andreas Fischbach, Martin Zaefferer, Jörg Stork, Martina Friese, Thomas Bartz-Beielstein

SPOTSeven Lab, Dept. of Comp. Sci. and Eng. Sci.
TH Köln
E-Mail: {andreas.fischbach, martin.zaefferer, joerg.stork, martina.friese, thomas.bartz-beielstein}@th-koeln.de
www.spotseven.de

1 Introduction

When researchers and practitioners in the field of computational intelligence are confronted with real-world problems, the question arises which method is the best to apply. Nowadays, there are several, well established test suites and well known artificial benchmark functions available. However, relevance and applicability of these methods to real-world problems remains an open question in many situations. Furthermore, the generalizability of these methods cannot be taken for granted. Some preliminary ideas about generalizability are discussed in [1, 2].

This paper describes a data-driven approach for the generation of test instances, based on real-world data, as depicted in Figure 1. The test instance generation uses data-preprocessing, feature extraction, modeling, and parameterization. It was applied to several real-world scenarios, e.g., in the context of genetic programming [3]. In this work we apply this concept on a classical design of experiment real-world project and generate test instances for benchmarking, i.e., design of experiment methods and model fitness. But it can also be used to compare and analyze several surrogate techniques and optimization algorithms as well.

In most cases, complex and expensive real-world problems do not provide sufficient data for comparison of methods. Thus, our goal is our goal is to create a toolbox containing multiple data sets of real-world projects. With that toolbox, researchers are granted access to both the data sets and the derived test functions.

This work mainly focuses on the following questions:

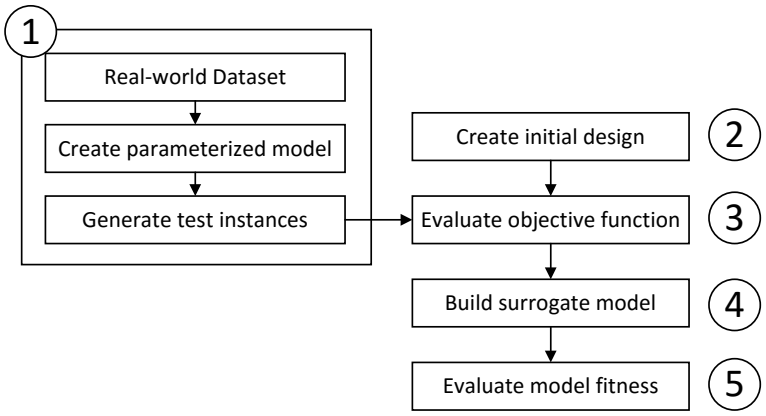

Figure 1: *Simplified test process of surrogate models fitted upon evaluation on initial designs with real-world data based test instances. An optional validation step can be added before the test instances are used in step 3.*

(Q-1) What is the characteristic of a variation of a certain model parameter in terms of the models fitness landscape?

(Q-2) How can similarity between models be computed and what are useful thresholds to separate similar models from almost equal models on the one hand, and completely different models on the other hand?

Considering an example application for the test function generator, the design of experiment used to gather the reference data set used in this work will be further analyzed:

(Q-3) Which design of experiment works best for the underlying real-world problem?

The remainder of this work is organized as follows. Section 2 gives a literature review. Section 3 describes the process of the test function generation and applied modeling techniques as well as model similarity measures used. Section 4 illustrates the reference data set and the example application, the evaluation of design of experiment methods. Section 5 concludes the work and gives a short outlook for future work.

2 Related Work

In the research field of benchmarking real-parameter problems many contributions address the random generation of problem instances based on user defined parameters, like the *Max-Set of Gaussian Landscape Generator* [4], or the *Krigifier* [5], both using Gaussian Processes.

The *Max-Set of Gaussian Landscape Generator* computes the upper envelope of m weighted Gaussian process realizations and can be used to generate continuous, bound-constrained optimization problems. The landscape generator is parameterized to control, e.g., the number of Gaussian components and implicitly, the number of local optima, the occurrences and variations of hills and the peaks, and the global optimum.

The *Krigifier* realizes a procedure for generating nonlinear objective functions. Its idea is based on the convenient supposition that objective functions are realizations of stochastic processes. The user specifies an underlying trend, a stochastic process and a finite number of points at which the process will be observed. The Krigifier creates a noise term and uses the trend and the noise term to produce an objective function.

A completely different approach is addressed by the *Real-Parameter Black-Box Optimization Benchmarking* [6]. The organizers of this benchmarking challenge choose and implement a benchmarking function testbed, typically covering artificial test functions like Sphere, Rosenbrock, Rastrigin, etc. Participants then have to apply their black-box problem solvers and their results will be gathered and compared.

All these approaches do not rely directly on real-world problems. Our work realizes one step towards closing a gap in means of validating methods regarding their applicability and generalizability in practical deployment.

3 Test Function Generation

The simplified process of real-world data based test function generation is depicted in Figure 2. It consists of the following steps:

1. Process a data set of a real-world problem, $\mathbf{X}$ is the design matrix and $\mathbf{Y}$ is a vector of corresponding outcome values of the underlying process.

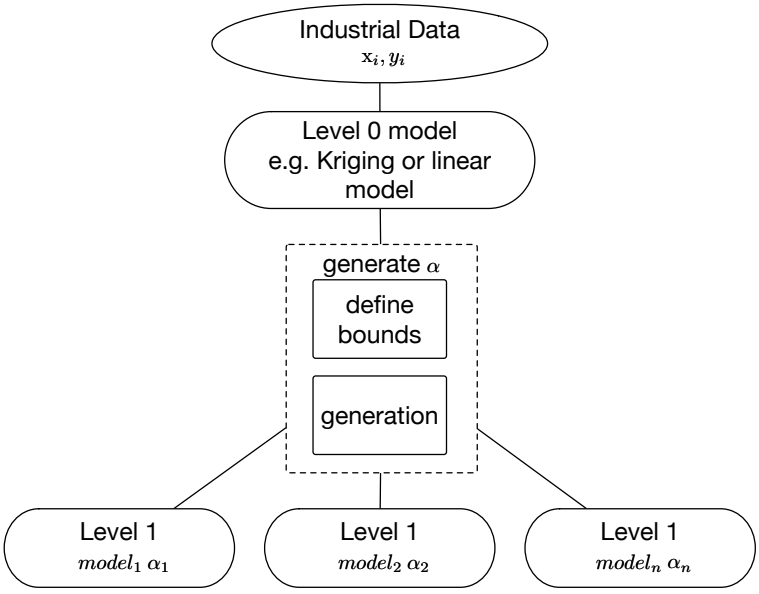

Figure 2: *Resulting hierarchy of the generation process of different models to be used as test function instances.*

2. Build a model, further denoted as *level 0 model*, that is suitable for regression and interpolation purposes of the data. In this work the Kriging technique is used to build the model.

3. Create a parameter α to vary the previously fitted *level 0 model*. The parameter α is a scalar or vector, that perturbs the generated model. It may, e.g., define a change in parameters or other variables of the derived model. First, bounds are generated for α, ensuring numerical robustness. Then, n instances α_i with $i = 1, ..., n$ are randomly created within the chosen bounds. Finally, a randomly selected subset of the desired number of $m < n$ instances is chosen. This ensures that the instances which need to be solved by the evaluated methods, are never known in advance.

4. Apply the selected α instances on the *level 0 model* to retrieve the desired number of m *level 1 model* instances, each coupled with an α_j, $j = 1, ..., m$. The process ensures the fulfillment of the requirements on the similarity demands of the models by computing similarity measures and discarding infeasible α instances.

The remainder of this section describes the Kriging modeling technique,

the model variation and the model similarity evaluation.

3.1 Kriging

Often used for the purpose of regression and interpolation, Kriging is a modeling method based on Gaussian processes. In the following we will stick closely to the descriptions by Forrester et al. [7]. Further details can be found in their book. Given a set of n solutions $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1\ldots n}$ in a k-dimensional continuous search space with observations $\mathbf{y} = \{y^{(i)}\}_{i=1\ldots n}$, Kriging tries to determine an expression for a predicted value at an unknown location by interpreting the observations $\mathbf{y}$ as realizations of a stochastic process. The stochastic process is defined by the set of random vectors $\mathbf{Y} = \{Y(\mathbf{x}^{(i)})\}_{i=1\ldots n}$. The correlation of the random variables $Y(\cdot)$ is modeled as follows [7]:

$$\operatorname{cor}\left[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})\right] = \exp\left(-\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j^{(l)}|^{p_j}\right). \tag{1}$$

The matrix that collects correlations of all pairs $\{(i,l)\}$ is called the correlation matrix $\mathbf{\Psi}$. It is used in the Kriging predictor

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \boldsymbol{\psi}^T \mathbf{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{2}$$

where $\hat{y}(\mathbf{x})$ is the predicted function value of a new sample $\mathbf{x}$, $\hat{\mu}$ is the maximum likelihood estimate (MLE) of the mean and $\boldsymbol{\psi}$ is the vector of correlations between training samples $\mathbf{X}$ and the new sample $\mathbf{x}$. The width parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_j, \ldots, \theta_k)^T$ determines how far the influence of each sample point $\mathbf{x}$ spreads. In detail, the larger the width parameter is, the faster are the potential changes in the predicted value. The smaller the width parameter is, the slower are the potential changes in the prediction. Since there is one θ_i for each dimension, this parameter can control the activity in each dimension. The parameter p_j is usually fixed at $p_j = 2$, and defines the shape of the correlation function: At $p_j = 2$, the correlation function is more smooth, whereas $p_j = 1$ is less smooth. In case of noise, the parameter λ is added to the diagonal of the correlation matrix $\mathbf{\Psi}$. This allows the model a more smooth fit through observations (regression), in contrast to the default which reproduces all training data exactly (interpolation). Classically, λ is used to deal with noisy data. But it can as well be used to smoothen more rugged fitness landscapes.

All model parameters are determined by Maximum Likelihood Estimation (MLE). For $\boldsymbol{\theta}, \lambda$ and $\boldsymbol{p}$, MLE requires numerical optimization.

3.2 Model variations

The main parameters controlling the behavior of the model are $\boldsymbol{\theta}$ and λ. The major goal of the test function generator is the deployment upon real-world data, which is usually noisy. So the variation of the parameter λ is a natural choice at a first glance. In addition the variation of the width parameter θ seems important to change the model (slightly) by maintaining the general characteristic of the fitness landscape under certain circumstances. The bounds of the variations of the parameters have to be defined carefully, otherwise the model can show signs of degeneration.

The test function generator will compute lower and upper bounds for α according to the fitted *level 0 model*. The first component of α represents the λ value and the remaining components represent the corresponding $\boldsymbol{\theta}$ values. They will be added to their corresponding values of the *level 0 model* to retrieve the altered *level 1 model*. Afterwards the correlation Matrix $\boldsymbol{\Psi}$ is recalculated, so that the changes take effect before predictions are made. The test function generation returns the desired number of similar functions, randomly drawn from the search space defined by the *level 0 model* and the bounds for α. If the given bounds do not allow the creation of sufficient feasible instances, an error message is produced.

An example is shown in Fig. 3. Here, the one-dimensional function

$$f(x) = (-18x - 2)^2 \sin(20x - 4)$$

is first sampled by 11 equidistant points. The derived level 0 Kriging model has $\lambda = 0$ and $\theta = 100$. To derive the level 1 models, the bounds for *alpha* are set to $alpha_{low} = [0, -90]$ and $alpha_{high} = [1, 900]$ so that λ will be set to values between zero and one, and θ between ten and one thousand. For demonstration purposes, the extreme values for *alpha* are chosen, as shown above each plot in Fig. 3. The plot shows that different α values affect the ruggedness of the function and vary the number of local optima.

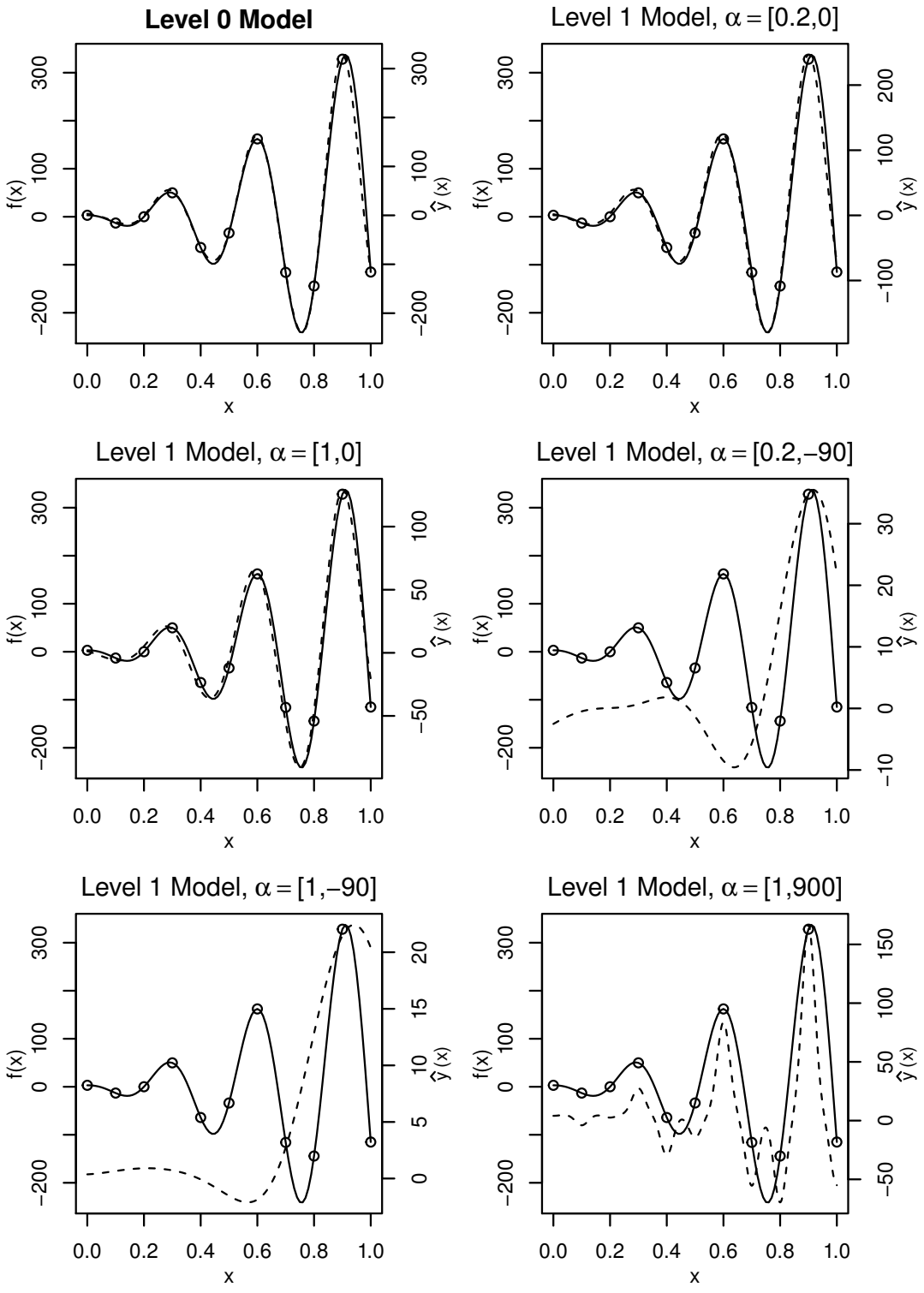

Figure 3: *A level 0 Kriging model and several derived level 1 models. The solid line is the true function $f(x)$, circles indicate observations used to fit the level 0 model and the dashed line indicates the level 1 model $\hat{y}(x)$.*

3.3 Model Similarity Computation

In addition to the quality of a model itself a measurement of the similarity of two models is needed. This should prevent the generation of test functions which are not anymore correlated to the given training data. Some correlation is desirable, so that the generated instances reflect the structure of the real-world data to some degree.

The *level 1 models* difference from the *level 0 model* will be computed evaluating the models at a pre defined equidistant grid and taking the differences of the predicted values of the models. It must be ensured that there is at least a difference large enough to ensure that the models are not equal or almost equal, otherwise we are always looking at the very same function. On the other hand, the difference must not be too large, because the characteristic of the underlying problem must be preserved, otherwise the results of the evaluations are questionable.

Each of the accuracy measures discussed in the following is unique and none can be considered as superior to the others. One goal in this work is to define an ensemble of measurements with carefully defined thresholds. This should lead to a confident set of different models comparable to the underlying real-world problem.

- The **mean absolute error** (MAE) is a popular measure to evaluate model fitness. It is defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

- The **root mean square error** (RMSE) has been used as a standard metric to measure model performance in different scientific fields like meteorology, air quality, and climate research studies [8]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|^2}{n}}$$

- **Pearsons correlation coefficient** (r) is a measure of the amount of linearity between two continuous variables X and Y, giving a value between -1 and $+1$ inclusive, where $+1$ is total positive correlation, 0 is no correlation, and -1 is total negative correlation. Regarding the prediction values of the *level 0 model* and *level 1 model*, correlation

values are desired to be closer to +1 as to 0. If they are too close to +1, they can be discarded due to their degree of conformity, while correlation values near 0 or even negative correlations indicates that the two models are not similar anymore.

- To be capable of dealing with non-linear models or non-linear changes of the models, rank based correlation is regarded as well. In this work **Spearmans** ρ rank correlation coefficient is used. An increasing coefficient value describes an agreement of the rankings. A value of $+1$ implies that the rankings are the same, -1 that the rankings are inverse and 0 that the rankings are independent.

- In addition a **t-test** will be performed to compare the two populations of predictions. The p-value of the t-test describes the likeliness of the observed data assuming the null hypothesis is true. The null hypothesis is that the compared data groups are from the same population. For two similar models a high p-value next to $+1$ is to be expected. It has to be regardet that t-test demands several assumptions, i.e., the normal distribution of the data.

Although these statistics have been used for years now, there is no consensus which of these statistics is superior. Measures of absolute values like the MAE or the RMSE lack interpretability, while statistics delivering values between -1 and $+1$ or 0 and $+1$ are easy to interpret if the distribution of the observed values is appropriate. As a conclusion, a combination of these statistics will be used.

The interface of the test function generator provides a function that takes training data, validation data, lower and upper bounds for the variation vector α and the number of test functions to generate. These instances are drawn randomly in the search space defined by the *level 0 model* and α. The test function generator performs the similarity computation automatically and discards generated models that are within the bounds for α but can not be seen as similar or are too similar according to the previously discussed statistics.

4 Example Application - DoE Evaluation

The test function generator was motivated by two different kinds of applications. The first application is to analyze the generalizability of a method developed for a specific problem. The second application is to further analyze a specific real world data set. The remainder of this section introduce the real-world data set, the experimental setup and the results of the experiments.

4.1 Reference Data Set

The industrial data, which depicts the basis for our test function generator, originated from experiments to optimize the development of gas sensors for combustion processes [9]. In a combustion process, a number of different gases are produced. A fraction out of these gases are crucial indicators for the goodness of the combustion and important for the process control. Therefore, the target of the new sensor design was to improve the sensor sensitivity for a specific gas, while reducing the cross sensitivity to other gases. The experiments were conducted using a mixture of seven different gases as input, further denoted as $X1$ to $X7$. Moreover a measurement in voltage of two different sensors were retrieved as output, further denoted as $Y1$ and $Y2$. The underlying experimental design was a response surface design fitted to built the associated regression model, including two-way interactions and quadratic behavior.

The design size was set to 80 experiments as a result of time-consuming real-world measurements. So, all experiments could be performed within one week. An overview of the data is outlined in Table 1. The data was anonymized and standardized due to confidentiality restrictions.

An additional 60 experiments were run, with the same characteristics as described above. The resulting dataset was used for model validation.

These two data sets were previously used to compare and analyze advantages and disadvantages of several different regression modeling techniques on sparse and limited data, e.g. Bayesian robust linear regression, standard regression approaches, like ordinary least squares and **Lasso** (least absolute shrinkage and selection operator, see [10]), and genetic programming [9]. The results show that the **Lasso** regression model performs best on

Table 1: *Overview of the standardized dataset used to generate the models of the sensors. Each input of the model is denoted by an X and each sensor output is denoted by an Y.*

	X1	**X2**	**X3**	**X4**	**X5**	**X6**	**X7**	**Y1**	**Y2**
Minimum	−1.13	−1.21	−1.16	−1.13	−1.15	−1.17	−1.00	−1.94	−2.06
1st Quartal	−1.13	−1.21	−1.16	−1.13	−1.15	−1.17	−0.82	−0.63	−0.58
Median	0.09	0.03	0.12	0.08	0.08	0.05	−0.39	0.06	0.09
Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3rd Quartal	1.30	1.26	1.40	1.29	1.28	1.28	0.59	0.66	0.67
Maximum	1.30	1.26	1.40	1.29	1.31	1.28	3.79	2.32	2.28

the validation data set predicting $Y1$. In the resulting model only the parameters $X1$ to $X4$ were included.

Due to the production process of the sensors, future experiments will most likely lead to similar linear regression models. The customer is heavily interested in these easy to understand models and from the research perspective it is very important and interesting to analyze the capabilities of the applied design of experiment. Therefore the test function generator will be set up to deliver similar problem instances based on the given data.

4.2 Experimental setup

As a first step to demonstrate the application of the test function generator, the variation of the Kriging parameters λ and θ will be analyzed. We use the data described in the previous section limited to the inputs $X1$ to $X4$, which are the most important and interesting parameters according to the results of the different modeling techniques [9].

The bounds for α for this screening experiment are given in Table 2. All experiments are performed using the free software environment for statistical computation, R[1]. As optimizer for the estimation of the Kriging model parameters θ and λ, Global Optimization by Differential Evolution (DEoptim), with a budget of 1000 evaluations is set.

To analyze the generalizability of the design of experiment created to obtain the training data, further referred to as *base design* we apply the

[1]R as well as all employed packages are available at `http://cran.r-project.org/`

Table 2: *Lower and upper bounds for λ and θ and the nr. of values (steps) taken in each dimension to build a grid as design of experiment to evaluate the influence of λ and θ variations on the model similarity.*

	lambda	**theta**
lower bound	0.1	-2
upper bound	1	10
steps	4	4

test function generator on the data set. The generator will be set up to generate modified Kriging models of the obtained *level 0 model* with previously adjusted similarity thresholds.

Several different designs will be employed and linear regression *level 2 models* fitted by means of the least squares method by evaluating the design points on the drawn *level 1 model* instances. The model fitness will be evaluated by computing the RMSE between the *level 1 models* and *level 2 models* at a predefined grid. These experiments are based on previous work to compare different design methods, e.g., random sampling, Latin hypercube sampling and full factorial designs, see also [11]. The test function generator is set up to generate 20 different *level 1 models*.

Several designs will be set up to evaluate the models. The *base design* will be pre processed, as the dispensable columns for the parameter $X5$ to $X7$ will be omitted and remaining duplicates will be removed. This results into a design size of 67 points. It has to be regarded that an optimized design created for the parameters $X1$ to $X4$ can look different.

This design will be compared with a Full factorial design (FFD) with 2 levels per factor (16 points) and a FFD with 3 levels per factor (81 points) as well. In addition Latin hypercube designs (Lhd) with 16, 67 and 81 points resp. and simple uniform random sampling designs of corresponding sizes will be created.

Each design will be used to retrieve predictions of the 20 different *level 1 models* and fit a linear regression model of the following form, which will be taken as a benchmark model structure:

$$\hat{y} = \beta_0 + \sum_{i=1}^{4} \beta_i x_i + \beta_{14} x_1 x_4 + \beta_{34} x_3 x_4.$$

The resulting models will be evaluated at a pre defined equidistant grid and the RMSE between the predictions of the *level 1 models* and the *level 2 models* will be computed.

4.3 Results

At first the results of the screening experiments of the variations of α will be discussed. In total, 1045 experiments with different α instances were run, resulting from the 4 different values per parameter (4^5 =1024) plus additional 20 runs of one factor at a time variations and one trial with each component in α set to 0 for validation purpose.

Results of the one factor at a time variations are shown exemplarily in Figures 4 and 5 for λ, θ_1 and θ_4. It can be seen that, even with increasing RMSE and MAE, the correlation coefficients seem insensitive against changes of one parameter, while the p value of the t-test seems very sensitive against changes. Another interesting aspect is, that the p value seems less sensitive for changes of λ than for the changes of the θ values.

In Figure 5, it can be seen, that the change of the variation of θ_4 from 2 to 4 only results in a small change of the RMSE and MAE. This sometimes can occur when changes of α are cut to the upper or lower bounds.

As expected, a linear model fit revealed that almost all components of α have a significant effect on the RMSE, see Table 3. Similar results were obtained for the other measures, i.e., MAE, Pearsons correlation coefficient r, Spearmans correlation and the p value of the t-test. The p value of the t-test seems to be the most sensitive of the measures.

Table 3: *Summary of a linear model fit, showing the estimated influence and significance of the different parameters on the RMSE.*

	Estimate	Std. Error	t value	p value	
(Intercept)	0.2029	0.0032	62.88	< 2e-16	***
lambda	0.1099	0.0047	23.32	<2e-16	***
theta1	-0.0013	0.0007	-1.83	0.0669	.
theta2	-0.0042	0.0007	-5.96	3.55e-09	***
theta3	-0.0140	0.0007	-19.95	<2e-16	***
theta4	-0.0093	0.0007	-13.33	<2e-16	***

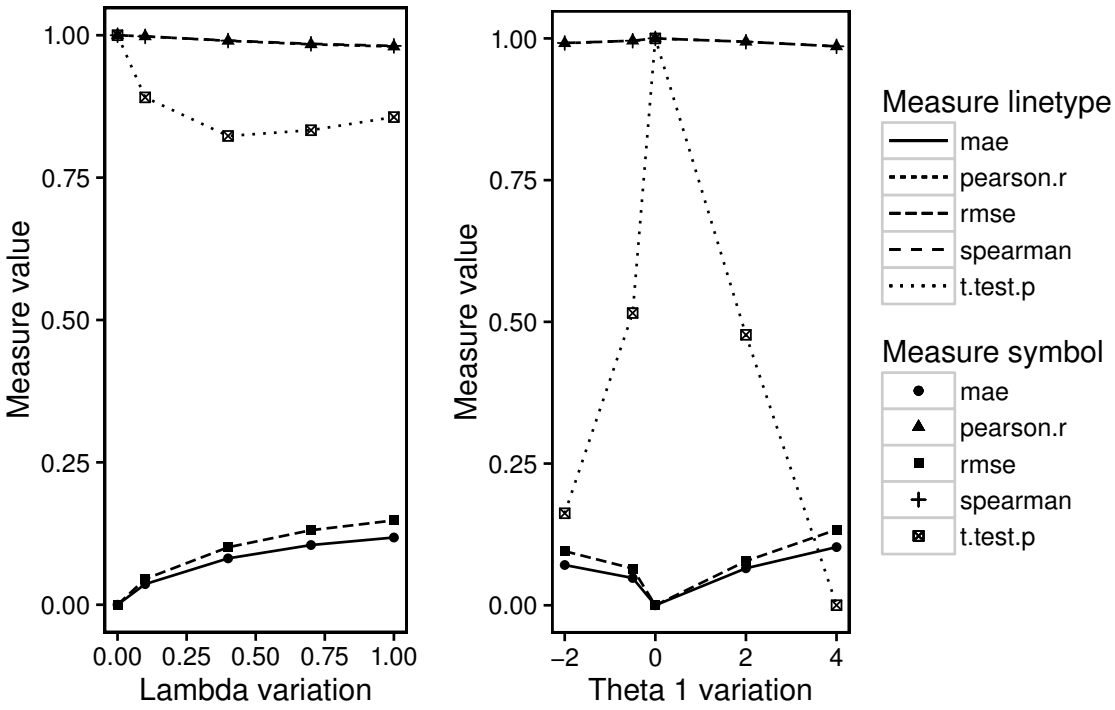

Figure 4: *Effects of the variation of model parameters on the chosen similarity measures MAE, RMSE, Pearsons r, Spearman and the t-test p value.* ***Left:*** *Effects of* λ *changes.* ***Right:*** *Effects of* θ_1 *changes.*

A detailed look at the result table from the screening experiment was taken, to decide which thresholds for each of the measures should be taken to discard test function instances. In the summary of the measures, depicted in Table 4, the distribution of the measures can be seen. The correlation coefficients does not become very low, with their mean at around 0.95. The p value of the t-tests distributes on the whole scale from 0 to +1.

Table 4: *Summary of statistical measures of similarity gathered on screening experiments for* α*.*

rmse	pearson.r	spearman	t.test.p
Min. :0.0000	Min. :0.8732	Min. :0.8560	Min. :0.0000006
1st Qu.:0.1819	1st Qu.:0.9268	1st Qu.:0.9283	1st Qu.:0.0189555
Median :0.2430	Median :0.9524	Median :0.9564	Median :0.1431840
Mean :0.2376	Mean :0.9486	Mean :0.9506	Mean :0.3314180
3rd Qu.:0.2935	3rd Qu.:0.9729	3rd Qu.:0.9749	3rd Qu.:0.6384715
Max. :0.4120	Max. :1.0000	Max. :1.0000	Max. :1.0000000

A full linear model with interaction terms between all parameters and main

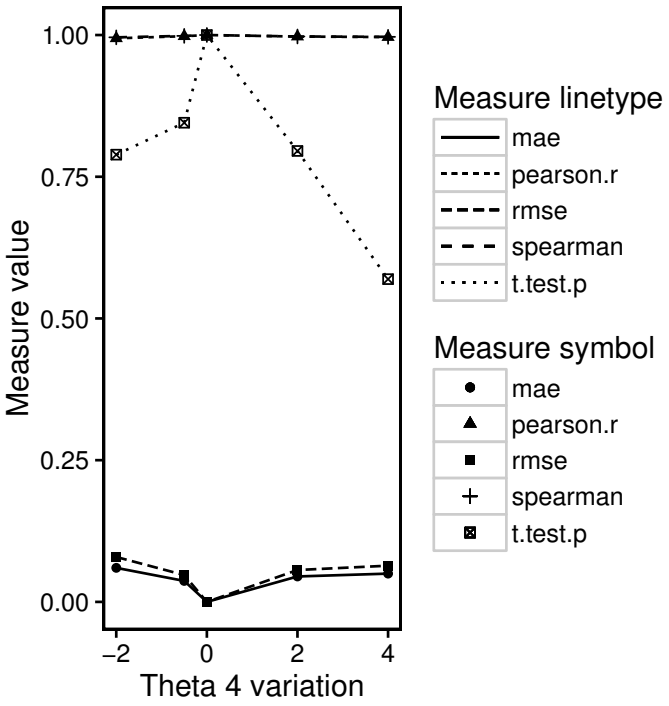

Figure 5: *Effects of the variation of θ_4.*

Table 5: *Thresholds for measures preventing too similar and degenerated test functions.*

	rmse	pearson.r	spearman	t.test.p
lower bound	0.2	0.85	0.85	0.5
upper bound	0.35	0.92	0.92	0.9

effects up to second degree predicting the RMSE and Spearman correlation coefficient reveals an adjusted coefficient of determination of about 0.85 and 0.74 respectively.

For the design of experiment application, the bounds for the measures were set es shown in Table 5. With these thresholds 20 random test functions were drawn. The impact of the chosen designs on the RMSE of the derived linear models compared to the test function are shown in Figure 6. It can be seen that the base design, applied to create the training data in the real-world project, could not be outperformed by any other design type. At a first glance it might be surprising, that designs with a larger design size, e.g. the *Uniform 81* or even the *LHD 81*, perform worse. But these designs were not set up to fit second order linear regression models and would therefore not be the first choice in such a setup. The results would surely

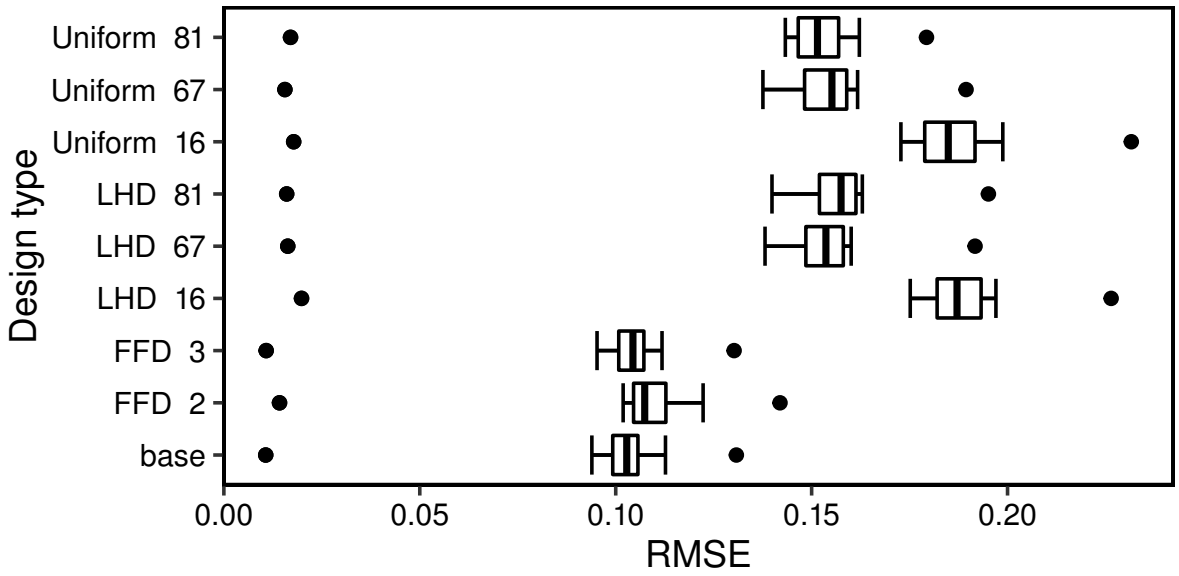

Figure 6: *Resulting RMSE values of models based on different design of experiment methods each applied on the same set of 20 random test functions.*

look different if the *level 2 model* would as well be a Kriging model.

5 Conclusion

In this work the generation process of test function instances based on a real-world industrial data set is described. Based on this data a Kriging model is fitted and altered according to a variation parameter α, by adding its components to the corresponding model parameter. The generated instances will be discarded or kept according to thresholds for several model similarity measures. This leads finally to a test function instance pool that can be used to benchmark, i.e, design of experiment and modeling methods, for their practical use on the underlying problem.

The major research Questions addressed in this work were:

(Q-1) **What is the characteristic of a variation of a certain model parameter in terms of the models fitness landscape?**
The variation of the Kriging model parameters λ and θ lead to altered models that were still correlated at rather high coefficient rates around 0.9. This was ensured by the definition of lower and upper bounds relative to the parameter values of the base *level 0*

model. This kind of limit the amount of changes applicable to the model and can be problematic for example when a large number of test instances is needed.

(Q-2) **How can similarity between models be computed and what are useful thresholds to separate similar models from almost equal models on the one hand, and completely different models on the other hand?**
This can kind of simple be visually analyzed for one or two dimensional problems, but is getting harder in general for larger number of dimensions. Statistical measures can help to distinguish models that are too similar or too different. Further analysis of the resulting fitness landscapes in sense of, e.g., number of local optima or gradients, can help to judge the similarity of two different models.

(Q-3) **Which design of experiment works best for the underlying real-world problem**
The application of different design of experiment methods has shown, that the base design, used to gather the training data set, was not outperformed by any other design method. Even larger designs like a Full factorial designs at three levels per factor and Latin hypercube designs with 81 points could not dominate the results. This of course has to be further analyzed by altering the model techniques for the *level 2 models*. Even slight changes to the base design can now be analyzed and might lead to interesting new design points for the customer in future real-world experiments.

Interesting future work include the shift of the similarity computation to the beginning of the instance generation. The Matrix computations to alter Kriging models can be regarded as time consuming, especially with high dimensional data. So it would be beneficial to compute an expectational value for the similarity of two models.

In addition, fitness landscape analysis is an interesting topic to include in the work. The fitness landscapes comparison based on the RMSE or correlations of function values can be extended by interesting features to add or to avoid, e.g., the number of local optima, plane areas or large gradients.

Finally, different variation methods on the models, e.g. rotating, scaling, distortion of the input space should be applied. Considering Kriging models, conditional simulation, could be applied to deliver possible realizations of a Gaussian process with a certain probability and therefore be very suitable

to deliver model variations based on real-world data.

References

[1] M. Chiarandini and Y. Goegebeur, "Mixed Models for the Analysis of Optimization Algorithms," in *Experimental Methods for the Analysis of Optimization Algorithms* (T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, eds.), pp. 225–264, Germany: Springer, 2010.

[2] T. Bartz-Beielstein, "How to Create Generalizable Results," in *Springer Handbook of Computational Intelligence*, pp. 1127–1142, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[3] O. Flasch, "A Modular Genetic Programming System," May 2015. Dissertation zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften der Technischen Universität Dortmund, Fakultät für Informatik.

[4] M. Gallagher and B. Yuan, "A general-purpose tunable landscape generator," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 590–603, Oct. 2006.

[5] M. W. Trosset, I. for Computer Applications in Science, and Engineering., *The Krigifier [microform] : a procedure for generating pseudorandom nonlinear objective functions for computational experimentation / Michael W. Trosset.* Institute for Computer Applications in Science and Engineering, NASA Langley Research Center ; National Technical Information Service, distributor Hampton, VA : Springfield, VA, 1999.

[6] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking 2009: Experimental setup," Tech. Rep. RR-6828, INRIA, 2009.

[7] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling.* Wiley, 2008.

[8] T. Chai and R. R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, pp. 1247–1250, June 2014.

[9] M. A. Rebolledo Coy, S. Krey, T. Bartz-Beielstein, O. Flasch, A. Fischbach, and J. Stork, "Modeling and Optimization of a Robust Gas Sensor," in *Bioinspired Optimization Methods and their Applications* (G. Papa and M. Mernik, eds.), pp. 267–278, May 2016.

[10] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B (Methodolgical)*, vol. 58, no. 1, pp. 267–288, 1996.

[11] A. Fischbach, J. Stork, M. Zaefferer, S. Krey, and T. Bartz-Beielstein, "Analyzing Capabilities of Latin Hypercube Designs Compared to Classical Experimental Design Methods ," in *25. Workshop Computational Intelligence* (F. Hoffmann and E. Hüllermeier, eds.), pp. 255–270, 2015.

Kontakt/Impressum

Diese Veröffentlichungen erscheinen im Rahmen der Schriftenreihe "CIplus". Alle Veröffentlichungen dieser Reihe können unter
https://cos.bibl.th-koeln.de/home
abgerufen werden.

Köln, Januar 2012

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.
Datum der Veröffentlichung: 14.11.2016

Herausgeber / Editorship

Prof. Dr. Thomas Bartz-Beielstein,
Prof. Dr. Wolfgang Konen,
Prof. Dr. Boris Naujoks,
Prof. Dr. Horst Stenzel
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
TH Köln,
Steinmüllerallee 1,
51643 Gummersbach
url: www.ciplus-research.de

Schriftleitung und Ansprechpartner/ Contact editor's office

Prof. Dr. Thomas Bartz-Beielstein,
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
TH Köln,
Steinmüllerallee 1, 51643 Gummersbach
phone: +49 2261 8196 6391
url: http://www.spotseven.de
eMail: thomas.bartz-beielstein@th-koeln.de

ISSN (online) 2194-2870

Technology
Arts Sciences
TH Köln

Technology
Arts Sciences
TH Köln