Surrogate Model for Mixed-Variables Evolutionary Optimization Based on GLM and RBF Networks

Lukáš Bajer[1,2] and Martin Holeňa[2]

[1] Faculty of Mathematics and Physics, Charles University
Malostranské nám. 25, Prague 1, Czech Republic
bajer@cs.cas.cz
[2] Institute of Computer Science, Czech Academy of Sciences
Pod Vodárenskou věží 2, Prague 8, Czech Republic
http://www2.cs.cas.cz/~martin/

Abstract. Approximation of costly objective functions by surrogate models is an increasingly popular method in many engineering optimization tasks. Surrogate models can substantially decrease the number of expensive experiments or simulations needed to achieve an optimal or near-optimal solution. In this paper, a novel surrogate model is presented. Compared to the most of the surrogate models reported in the literature, it has an advantage of explicitly dealing with mixed continuous and discrete variables. The model use radial basis function networks for continuous and clustering and a generalized linear model for the discrete covariates. The applicability of the model is shown on a benchmark problem, and the model's regression performance is further measured on a dataset from a real-world application.

Key words: surrogate modelling, RBF networks, genetic algorithms, mixed-variables optimization, continuous and discrete variables

Author's preprint. The original publication is available at www.springerlink.com.
DOI: 10.1007/978-3-642-35843-2_41

1 Introduction

Different kinds of optimization tasks are encountered in many of todays engineering or industrial applications. Frequently, they are characterized by a high number of both continuous and discrete variables [1,2]. Such tasks are called *mixed-variables optimization problems*, often abbreviated as MVOP, and they are similar to *mixed-variable non-linear programming* (MINLP) tasks. These tasks are common especially in situations where the value of the objective function is obtained through some measurement, experiment or simulation.

A popular approach to tasks with costly objective functions is substituting an approximating model for the empirical objective function. This approach,

called surrogate modelling [3,4], is widely used in connection with evolutionary algorithms (EAs), in spite of having been originally introduced in the area of smooth optimization.

Assessing some of the individuals with not necessary accurate, but much faster model brings an important benefit: a notably larger population can be evolved in parallel. Even though the original fitness function can be evaluated only on a limited number of individuals, the EA can explore a larger part of the input space.

This paper describes a particular surrogate model based on radial basis function (RBF) networks. As far as we know, the existing publications about surrogate modelling in evolutionary optimization deal with only continuous domains or combination with integer variables [5]. Outside the evolutionary area, the work of Holmström [6] is known, for example. Interesting applications can be found in some articles [7,8,9], or SO-MI algorithm has appeared recently [10]. However, we are aware of no surrogate models for MVOP in evolutionary context. And even outside that context, such models are rather few.

In our model, multiple RBF networks are first trained on the continuous part of the data – in this phase, discrete variables are used to focus training of the networks on the most appropriate data. Next, a generalized linear model is trained taking discrete variables as the independent and residuals of the RBF networks as the dependent variables for training.

The paper is organized as follows: in the next section, we recall the optimization task and GLM principles. Section 3 describes our approach to constructing a surrogate model and using it in optimization. Finally, Section 4 provides the results of testing on a benchmark function and real-world data.

2 Optimization Task and Involved Methods

For any given objective function $f : \mathbf{S} \to \mathbb{R}$, we consider the *mixed-variable optimization problem* as finding x^* such that

$$f(\mathbf{x}^\star) = \max_{\mathbf{x}\in\mathbf{S}} f(\mathbf{x}). \tag{1}$$

where $\mathbf{x}^\star = (x_1^{(\mathrm{C})}, \ldots, x_n^{(\mathrm{C})}, x_1^{(\mathrm{D})}, \ldots, x_d^{(\mathrm{D})}) \in \mathbf{S}$. The problem includes n continuous and d discrete variables; their values belong to corresponding subspaces $\mathbf{S}^{(\mathrm{C})}$ and $\mathbf{S}^{(\mathrm{D})}$. This holds for maximization problem, minimization can be defined analogously. In addition, we suppose that the value sets $V_s(X_i^{(\mathrm{D})}),\ i = 1, \ldots, d$ of the discrete variables are finite and we do not differentiate between ordinal or nominal categorical variables – we do not require any ordering on any of the $V_s(X_i^{(\mathrm{D})})$.

2.1 Involved Methods

Evolutionary optimization, RBF networks and surrogate modelling were recalled already in the preceding paper [11]. For a more detailed treatment of them,

the reader is referred to specialized monographs, in particular to the recent monograph [12] for global genetic optimization, [13] and [14] for the traditional linear and polynomial response-surface models, which inspired the modern nonlinear surrogate models such as Gaussian processes and artificial neural networks [3,15,16].

Generalized Linear Models. The new surrogate model presented in section 3 relies in addition on *generalized linear models* (GLM). A GLM is a natural generalization of a linear regression model [17]. It consists of three parts: (1) the *random component* – observed values $\mathbf{Y}$ following a distribution from an exponential family with mean $E(\mathbf{Y}) = \boldsymbol{\mu}$ and constant variance σ^2, (2) the *systematic component* which relates values of explanatory (input) variables $(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^d)$ through a linear model with parameters $\beta_1, \ldots, \beta_d$ to a *linear predictor* $\boldsymbol{\eta}$

$$\boldsymbol{\eta} = \sum_{j=0}^{d} \mathbf{x}^j \beta_j, \tag{2}$$

and (3) a *link function* g that connects the random and systematic components together: $\boldsymbol{\eta} = g(\boldsymbol{\mu})$.

3 Proposed Surrogate Model for MVOP Evolutionary Optimization

In the preceding paper [11], we described a surrogate-assisted genetic algorithm (GA), in which the proposed surrogate model obtained in the following steps:

1. the training data is clustered according to $\mathbf{S}^{(\mathrm{D})}$,
2. for each cluster, separate RBF network is trained on $\mathbf{S}^{(\mathrm{C})}$.

Let us call this model *RBF/discrete clustering*, or shortly *DSCL*.

The DSCL model does not use any relationships of the discrete input variables to the response variable. That motivated further surrogate models for MVOP, which already use such relationships. A new version of a model combining RBF networks and GLM, called a RESID model, is introduced in the following text.

3.1 Model Construction

Construction of the RESID model starts with building previously described DSCL model $\hat{f}_{\mathrm{DSCL}} : \mathbf{S} \to \mathbb{R}$ which is composed of several RBF networks fitted on the continuous input variables on respective clusters from clustering made using the discrete variables. If different combinations of values of the discrete variables are possible, then different RBF networks on $\mathbf{S}^{(\mathrm{C})}$ need to be constructed to reflect the differences in those combinations. In fact, in the rare situation that there are enough training data to construct separate networks for all combinations of values of discrete variables, DSCL constructs all those separate networks.

In the next step, residuals of this DSCL submodel are computed for all N available training data

$$\delta_i = y_i - \hat{f}_{\mathrm{DSCL}}(\mathbf{x}_i^{(\mathrm{C})}, \mathbf{x}_i^{(\mathrm{D})}), \quad i = 1, \ldots, N. \tag{3}$$

Further, a GLM is fitted taking discrete variables as independent and these δ_i's as a dependent variable. The final value of the RESID model then sums the resulting values of DSCL and GLM models.

A pseudo-code of our algorithm for the RESID model fitting is given in Fig. 1. Important parts of the algorithm are briefly explained below.

3.2 Fitting the RESID Model

RBF networks enable us to use only continuous variables for their fitting. Construction of the DSCL submodel starts with clustering of the available training data according to their discrete values into several clusters in order to focus the RBF networks training on the most similar data points. Subsequently, separate networks are fitted with the data of each such a cluster using the data points' continuous variables.

Further, relation between values of the discrete input variables and the response variable is utilized via GLM which form an important part of our new RESID surrogate model.

Parameters of the DSCL Model. The sizes of the clusters $C_j, j = 1, \ldots, m$ have to fulfill

$$|C_j| \geq s_{\min}^{(g_{\min})} \tag{4}$$

where $s_{\min}^{(g)}$ is the minimal number of data needed for fitting an RBF network with g components, and $g_{\min}$ is the required minimal number of components of each of the fitted networks (implicitly, $g_{\min} = 1$, but the user can increase this value). The value $s_{\min}^{(g_{\min})}$ depends on the employed radial basis functions, and on the way of estimating the fitting errors $e(j, g)$ of RBF networks In particular, for Gaussian networks with d-dimensional continuous inputs and $e(j, g)$ estimated using k-fold cross-validation:

$$s_{\min}^{(g)} = \begin{cases} g\frac{k}{k-1}(d+2) & \text{if uncorrelated cont. variables with identical variance,} \\ g\frac{k}{k-1}(2d+1) & \text{if uncorrelated cont. variables with diagonal variance,} \\ g\frac{k}{k-1}\frac{d(d+3)+1}{2} & \text{else.} \end{cases} \tag{5}$$

If $e(j, g)$ are estimated without cross-validation, the coefficient $\frac{k}{k-1}$ is left out. One separate RBF network rbf_j is trained on the data of each cluster C_j, $j = 1, \ldots, m$. The maximal number of components of each network is upper-bounded by

$$g_j^{\max} = \lfloor (\frac{k-1}{k}|C_j|)/s_{\min}^{(g_{\min})} \rfloor. \tag{6}$$

GLM Model. Generalized linear model is used in its continuous-response form. In the constructed GLM, the response's distribution is assumed to belong to an exponential family: either normal, gamma or inverse Gaussian distribution is supported. The most proper distribution is chosen through cross-validation. Further, the corresponding link function for each of the three distributions is used.

FitTheModel($s^{(1)\min}$, **D**, e)
Arguments: $s^{(1)\min}$ – min. size of clusters,
D – training data, e – type of error estimate:
MSE, AIC, or BIC
Steps of the procedure:
(1) $\{C_j\}_{j=1}^{m} \leftarrow$ cluster **D** into clusters of size
at least $s^{(1)\min}$ according to discrete variables
(2) **for** each cluster C_j, $j = 1, \ldots, m$
(3) $rbf_j \leftarrow$ parameters of the RBF network with $g_j^\star$ components fitted with data from C_j, and with error e_j
(4) $e_j \leftarrow e[j, g_j^\star]$
(5) $\delta_i \leftarrow (y_i - \hat{f}_{\mathrm{DSCL}}(\mathrm{x}_i^{(C)}, \mathrm{x}_i^{(D)}; rbf))$ for $i = 1, \ldots, N$
(10) $glm \leftarrow$ fit the GLM on $(\delta_i)_{i=1}^{N}$
choose the best link function g and coding
of variables via cross-validation
Output: $\{(rbf_j, e_j)_{j=1}^{m}, glm\}$

Fig. 1. Pseudo-code of the fitting procedure

Before using or fitting the GLM, the discrete values must be converted to a proper representation. Since we do not assume any ordering of the discrete values, our default choice is *dummy coding* [17] which establishes one binary indicating variable $I_{ij} \in \{0, 1\}$ for each nominal value from the value sets $V_s(X_i^{(\mathrm{D})})$, $i = 1, \ldots, d$, $j = 1, \ldots, |V_s(X_i^{(\mathrm{D})})|$ of the original discrete variables.

Needless to say, the number of GLM parameters for GLM fitting can grow rapidly which restricts the applicability of the dummy coding, moreover default integer representation with ordering taken from the original integer values can be beneficial to the regression quality. Therefore, as an alternative representation, integer coding can be used in situation when there are not enough data for dummy coding. As default, coding resulting in a higher regression quality in terms of cross-validation error is used.

3.3 Evaluation with the Surrogate Model

Once the surrogate model is built, it can be used for evaluating individuals resulting from the evolution. The parameters of the model can be summarized

as $\{(rbf_j, e_j)_{j=1}^m, glm\}$. Here, rbf_j are RBF/DSCL network parameters, e_j are errors obtained from cross-validation, and $glm = (\beta_0, \ldots, \beta_r; \text{coding}, g)$ are parameters of the GLM.

Given a new individual $(\mathbf{x}^{(\mathrm{C})}, \mathbf{x}^{(\mathrm{D})})$, evaluation with the surrogate model starts with finding the index c of the cluster with the data closest to the individual's discrete values

$$c = \arg \min_{j=1,\ldots,m} \frac{1}{|\mathbf{N}_j|} \sum_{\mathbf{y} \in \mathbf{N}_j} d_{\mathrm{DISCR}}(\mathbf{x}^{(\mathrm{D})}, \mathbf{y}). \tag{7}$$

Here, d_{DISCR} denotes Hamming (used in testing) or Jaccard metric. Then, the RBF network with parameters rbf_c corresponding to this cluster c is used as a surrogate model of the original fitness by computing its return value on continuous dimensions $y_{\mathrm{DSCL}} = \hat{f}_{\mathrm{DSCL}}(\mathbf{x}^{(\mathrm{C})}; rbf_c)$. If more than one cluster is at the same distance from the individual, the RBF network with the lowest error e is chosen.

The resulting value is obtained by summing with the GLM response on the discrete variables

$$\hat{y} = y_{\mathrm{DSCL}} + g^{-1}\left(\sum_{j=0}^{d} x^{j(\mathrm{D})} \beta_j\right) \tag{8}$$

where g^{-1} is an inverse of the link function chosen during GLM fitting.

4 Implementation and Results of Testing

Our algorithms were implemented in the MATLAB environment. We utilized the Global Optimization Toolbox whose modified genetic algorithm was used as a platform for testing the model on a benchmark optimization task. Similarly, clustering method extends the standard bottom-up hierarchical cluster analysis from the Statistical Toolbox in order to guarantee clusters of a minimum specified size. Statistical Toolbox provide us with GLM fitting procedure, too, and we employ a nonlinear curve-fitting from the Optimization Toolbox for fitting RBF networks.

4.1 Model Fitting

Our models have been tested on two different kinds of data. The first, real-world dataset is the same as in our preceding articles [11,18], and comes from optimization in the domain of chemical catalysis. In the latter article [18], more information about this specific real-world application is provided as well as one particular boosted surrogate model based on multilayer perceptrons. The second dataset is a set of individuals resulted from one GA optimization run of the modified Schwefel's benchmark function [19].

The first dataset comes from a real application in chemical engineering – optimization of chemical catalysts for Hydrocyanic acid (HCN) synthesis [20].

Solutions of this task are composed of two discrete and 11 continuous variables, the whole dataset has 696 items. The dataset was randomly split into training (556) and testing (140) part.

As the second dataset, individuals from the first 10 generations of a run of GA optimization of modified Schwefel's function were taken. The original Schwefel's function $y = \frac{1}{p}\sum_{i=1}^{p} -x_i \sin(\sqrt{|x|})$ was modified in order to be defined on both continuous $(x_1^{(\mathrm{C})}, \ldots, x_p^{(\mathrm{C})}) \in [-512{,}512]^p$ and discrete $(x_1^{(\mathrm{D})}, \ldots, x_p^{(\mathrm{D})}) \in \{-10, -9, \ldots, 10\}$ variables in the following form

$$\begin{aligned} y \quad = \quad & \frac{1}{p}\sum_{i=1}^{p} -x_i^{(\mathrm{C})} \sin(\sqrt{|x_i^{(\mathrm{C})}|}) + \frac{10}{p}\sum_{i=1}^{p} \sin(\sqrt{|\pi(x_i^{(\mathrm{D})}) .* x_i^{(\mathrm{C})}|}) \\ & + \frac{20}{p}\sum_{i=1}^{p} \pi(x_i^{(\mathrm{D})})^2 \end{aligned} \tag{9}$$

where $\pi : \{-10, \ldots, 10\} \to \{-10, \ldots, 10\}$ is a random permutation. For simplicity, we have chosen only two continuous and two discrete variables ($p = 2$); the training set has 681 and testing set 101 data.

Our models (RESID and former DSCL) were compared with the RBF network from SUMO toolbox [15] using approximately the same computational time. Each of the three models (DSCL, RESID and SUMO) were 50 times trained on the training sets of the two tasks.

Results in Table 1 show that the new RESID model achieves about 10% lower root-mean-square error (RMSE) than the former DSCL model; the improvement is statistically significant (one-sided Mann–Whitney U test, $p_{\mathrm{HCN}} = 2.28 \cdot 10^{-12}$, $p_{\mathrm{Schwefel}} = 1.11 \cdot 10^{-10}$). The testing errors of both our models are considerably lower than errors of RBF networks from SUMO toolbox. Moreover, in the case of the HCN real-world dataset, the improvement of RESID model is even more than 25% compared to the results in [11]. Even though SUMO toolbox's training error is low, very large testing error shows poor generalization capabilities; see also Fig. 4.1.

Table 1. Surrogate-models' regression results on HCN and Schwefel dataset, average results from 50 trainings. RMSE on the testing and training set is supplemented by average medians of residuals on the testing set

dataset	model	RMSE (test set)	RMSE (train set)	medians of residuals
HCN	DSCL	8.2918 ± 0.4373	7.6337	4.9433
	RESID	7.6212 ± 0.3187	6.6131	4.0678
	SUMO	115.196 ± 34.1387	0.5711	35.0051
Schwefel	DSCL	62.712 ± 3.4248	83.553	4.955
	RESID	55.605 ± 5.4867	73.468	5.411
	SUMO	$64576 \pm 1.9\mathrm{e}{+05}$	33.964	44048

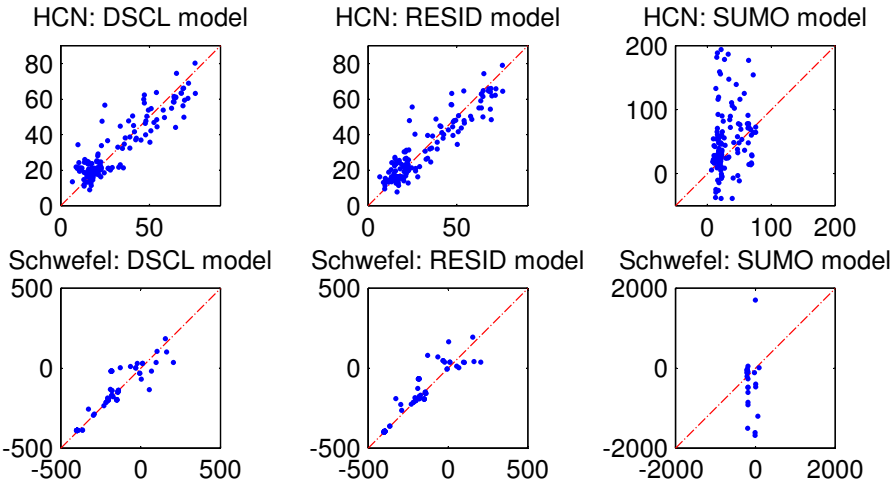

Fig. 2. Scatter plots of the DSCL, RESID and SUMO's RBF models on testing data

4.2 Genetic Algorithm Performance on the Benchmark Fitness

The Schwefel's benchmark fitness enabled us to test the model as a surrogate model for genetic optimization. The parameters of the function, especially the number of variables, was the same as in the case of the regression test.

Crucial criterion of successful optimization of empirical functions is the number of original fitness evaluations needed for reaching sufficient near-optimal solution. Therefore, the number of original fitness evaluations was measured in each of 100 runs in the moment when the following thresholds were reached: 1.2-, 1.1-, 1.05-, 1.02- and 1.01- multiple of the global minimum (corresponding to reaching value 20, 10, 5, 2 and 1 per cent above optimal solution). The numbers of needed evaluations for these limits are shown in Fig. 4.2. Since not every single run converged to all of these values, the numbers of evaluations belonging to each threshold were divided by the ratio of runs successfully reaching respective thresholds.

The average number of evaluations needed to get 1%-above-optimal solution has been significantly decreased from 13224 by more than 35% to 8310 with our RESID model ($p = 0.011$, one-sided Mann–Whitney U test), and by nearly 25% (to 10014) with the DSCL model; this result is not significant, though. RBF networks from the SUMO toolbox did not improve results of this optimization task (the average number of original evaluations was 13249).

5 Conclusion

This paper presented a novel kind of surrogate model for mixed-variable continuous and discrete optimization. It utilizes a clustered model with RBF networks

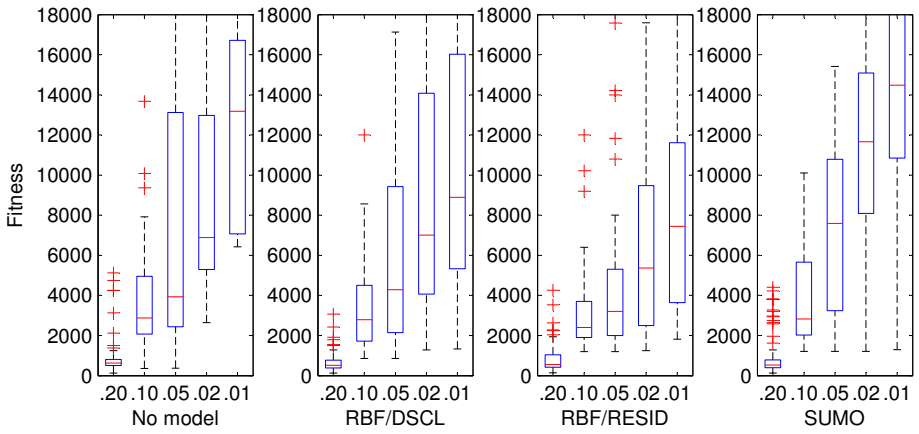

Fig. 3. The numbers of original evaluations needed to reach 1.2-, 1.1-, 1.05-, 1.02- and 1.01-multiple of minimum, measured on 100 GA runs

(one network per cluster) and a generalized linear model. The model focuses training of the RBF networks using clustering on the discrete part of the data. Generalized linear model trained on the discrete input variables further improves regression capabilities. Results of testing on two different datasets showed that the model is a competitive kind of regression for costly objective functions. Using this surrogate model for the optimization of modified Schwefel's benchmark fitness function resulted in saving up to 35 per cent of the original evaluations.

Acknowledgements This work was supported by the Grant Agency of the Charles University (GA UK), grant number 278511/2011, and by the Czech Science Foundation (GA CR), grant numbers P202/11/1368 and 201/08/0802.

References

1. Olhofer, M., Arima, T., Sendhoff, T.S.B., Japan, G.: Optimisation of a Stator Blade Used in a Transonic Compressor Cascade with Evolution Strategies. Evolutionary Design and Manufacture (2000) 45–54
2. Holeňa, M., Cukic, T., Rodemerck, U., Linke, D.: Optimization of catalysts using specific, description-based genetic algorithms. Journal of chemical information and modeling **48**(2) (2008) 274–282
3. Booker, A., Dennis, J., Frank, P., Serafini, D., V., T., Trosset, M.: A rigorous framework for optimization by surrogates. Structural and Multidisciplinary Optimization **17** (1999) 1–13
4. Gorissen, D., Dhaene, T., DeTurck, F.: Evolutionary model type selection for global surrogate modeling. Journal of Machine Learning Research **10** (2009) 2039–2078
5. Hemker, T., Fowler, K., Farthing, M., von Stryk, O.: A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. Optimization and Engineering **9**(4) (2008) 341–360

6. Holmström, K., Quttineh, N.H., Edvall, M.: An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. Optimization and Engineering **9**(4) (2008) 311–339
7. Abramson, M.A.: Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. Optimization and Engineering **5**(2) (2004) 157–177
8. Younis, A., Dong, Z.: Global optimization using mixed surrogate models for computation intensive designs. In: 2nd International Symposium on Computational Mechanics Hong Kong. Volume 1233. (2010) 1600–1605
9. Singh, G., Grandhi, R.V.: Mixed-Variable optimization strategy employing multifidelity simulation and surrogate models. AIAA Journal **48**(1) (2010) 215–223
10. Müller, J., Shoemaker, C.A., Piché, R.: SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Computers & Operations Research **In Press** (2012)
11. Bajer, L., Holeňa, M.: Surrogate model for continuous and discrete genetic optimization based on RBF networks. In: Intelligent Data Engineering and Automated Learning – IDEAL 2010. Volume 6283 of LNCS., Springer (2010) 251–258
12. Schaefer, R., Telega, H.: Foundations of global genetic optimization. Springer (2007)
13. Buhmann, M.D.: Radial basis functions: theory and implementations. Cambridge Univ. Press (2003)
14. Myers, R.H., Montgomery, D.C., Anderson-Cook, C.M.: Response surface methodology: process and product optimization using designed experiments. Volume 705. John Wiley & Sons Inc (2009)
15. Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., Crombecq, K.: A surrogate modeling and adaptive sampling toolbox for computer based design. Journal of Machine Learning Research **11** (2010) 2051–2055
16. Buche, D., Schraudolph, N., Koumoutsakos, P.: Accelerating evolutionary algorithms with gaussian process fitness function models. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews **35**(2) (2005) 183–194
17. McCullagh, P., Nelder, J.A.: Generalized linear models. Chapman & Hall (1989)
18. Holeňa, M., Linke, D., Rodemerck, U., Bajer, L.: Neural networks as surrogate models for measurements in optimization algorithms. In Al-Begain, K., Fiems, D., Knottenbelt, W., eds.: Analytical and Stochastic Modeling Techniques and Applications. Volume 6148 of LNCS., Springer (2010) 351–366
19. Schwefel, H.: Numerische Optimierung Von Computer-Modellen Mittels Der Evolutionsstrategie: Mit Einer Vergleichenden Einführung in Die Hill-Climbing-und Zufallsstrategie. Birkhäuser (1977)
20. Möhmel, S., Steinfeldt, N., Endgelschalt, S.., Holeňa, M., Kolf, S., Dingerdissen, U., Wolf, D., Weber, R., Bewersdorf, M.: New catalytic materials for the high-temperature synthesis of hydrocyanic acid from methane and ammonia by high-throughput approach. Applied Catalysis A: General **334** (2008) 73–83