RBF-based surrogate model for evolutionary optimization*

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, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, Prague 8, Czech Republic
martin@cs.cas.cz

Abstract. *Many today's engineering tasks use approximation of their expensive objective function. Surrogate models, which are frequently used for this purpose, can save significant costs by substituting some of the experimental evaluations or simulations needed to achieve an optimal or near-optimal solution. This paper presents a surrogate model based on RBF networks. In contrast to the most of the surrogate models in the current literature, it can be directly used for problems with mixed continuous and discrete variables – clustering and generalized linear models are employed for dealing with discrete covariates. The model has been tested on a benchmark optimization problem and its approximation properties are presented on a real-world application data.*

1 Introduction

Optimization of different kinds of empirical objective functions is included in many of todays engineering or industrial applications – in situations where the value of the objective function is obtained through some measurement, experiment or simulation. High costs or extensive time demands needed for evaluating such functions motivate engineers to reduce the number of such evaluations.

Surrogate modelling [3, 6] is a popular approach which substitutes an approximating model for some of the original function evaluations. This concept is widely used in connection with evolutionary algorithms (EAs). Here, some of the individuals are assessed with not necessary accurate, but much faster model. This brings an important benefit: a notably larger population can be evolved in parallel. Even though the precise evaluation can be made only on a limited number of individuals, the EA can explore a larger part of the input space.

Lots of current literature covers optimization in continuous, or in discrete domains. However, the area of industrial optimization is often characterized by both continuous and discrete variables [16, 7]. This paper describes a particular surrogate model based on radial basis function (RBF) networks and generalized linear models (GLMs). Most of the existing works [22, 17, 9] deal with only continuous domains or combination with integer variables, but the works dealing with mixed-variables surrogate models are rather few [21, 19].

In our model, multiple RBF networks are trained and discrete variables are used either for focusing training of the networks on the most appropriate data, or generalized linear model is constructed on this part of the data.

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

2 Problem description

For any given objective function $f : \mathbf{S} \to \mathbb{R}$, we consider the *mixed-variable optimization problem* (maximization) as finding the global optimum $\mathbf{x}^\star = (x_1^{(\mathrm{C})}, \ldots, x_n^{(\mathrm{C})}, x_1^{(\mathrm{D})}, \ldots, x_d^{(\mathrm{D})}) \in \mathbf{S}$ such that

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

The search space $\mathbf{S}$ has of n continuous and d discrete variables; forming corresponding subspaces $\mathbf{S}^{(\mathrm{C})}$ and $\mathbf{S}^{(\mathrm{D})}$. 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 distinguish between ordinal or nominal categorical variables – we assume no ordering on any of the $V_s(X_i^{(\mathrm{D})})$.

2.1 Involved methods

Surrogate modelling. Approximation of the fitness function with some regression model is a common cure for tasks when empirical objective function has to be used. These *surrogate models* simulate behaviour of

* This work was supported by the Grant Agency of the Charles University (GA UK), grant number 278511/2011 (Lukáš Bajer), and by the Czech Science Foundation (GA CR), grant number 201/08/0802 (Martin Holeňa).

the original function while being much cheaper and much less time consuming to evaluate.

As a surrogate model, mainly nonlinear regression models are used, for example gaussian processes [4] or artificial neural networks. In connection with evolutionary optimization, neural networks of the type multilayer perceptrons [10] and networks with radial basis functions [22, 17] have been particularly popular. The last mentioned kind of neural networks underlies also the model reported in this paper.

Combining of the original fitness function and the surrogate model is determined by *evolution control* (EC). In the literature [10], individual and generation based approaches are distinguished. While the individual-based EC chooses for evaluation by the original fitness only part of an enlarged population, the generation-based approach evaluates in different generations the whole population by either the original, or the model fitness.

RBF networks compute a mapping from the input space (typically a subspace of $\mathbb{R}^n$) to $\mathbb{R}$ (for simplicity we will focus on versions with scalar output) [5]. The mapping can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{g} \pi_i f_i(||\mathbf{x} - \mathbf{c}_i||) \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input, $g \in \mathbb{N}$ the number of components, $f_i : \mathbb{R}^n \to \mathbb{R}$ are radial basis functions, $\pi_i \in \mathbb{R}$ their weights, $\mathbf{c}_i \in \mathbb{R}^n$ radial functions' centres, and $||.||$ is a norm. As functions f_i, Gaussian functions with scalar width δ_i and euclidean norm $f_i(\mathbf{x}; \mathbf{c}_i, \delta_i) = e^{-\delta_i ||\mathbf{x}-\mathbf{c}_i||^2}$ are used most commonly.

Generalized linear models are a natural generalization of classical linear regression models [13]. They consist of three parts: (1) the *random component* – independent observed values $\mathbf{Y}$ following a distribution from the 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 $\beta_1, \ldots, \beta_d$

$$\boldsymbol{\eta} = \sum_{j=0}^{d} \mathbf{x}^j \beta_j \tag{3}$$

to a *linear predictor* $\boldsymbol{\eta}$, and (3) a *link function* g that connects the random and systematic components together: $\boldsymbol{\eta} = g(\boldsymbol{\mu})$. The explanatory variables are usually supplemented with the constant vector of ones corresponding to an intercept parameter β_0.

GLMs are particularly useful for our work because they are able to express a relation between discrete (integer or after a transformation of values even nominal) input variables and a continuous response.

3 Our strategy for using surrogate-assisted genetic optimization

Our version of the surrogate-assisted genetic algorithm including a detailed pseudo-code has been introduced in the previous article [1]. This section describes the construction and using of surrogate models based on RBF networks.

3.1 Model construction

RBF networks, which were defined in Section 2.1, enable us to use only continuous variables for their fitting. Construction of our first surrogate model [1] 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 datapoints. Let us call this model *RBF/discrete clustering*, or shortly *RBF/DSCL model.* Subsequently, separate RBF networks are fitted with the data of each such a cluster using the datapoints' continuous variables. The algorithm is the same as described on the Fig. 1 except the omitted steps (1)–(3), and the clustering which is made using discrete values from the training database $\mathbf{D}$ in the step (4).

This approach does not utilize relation between values of the discrete input variables and the response variable. As was stated in Section 2.1, such a relation can be expressed by generalized linear models, and these models form an important part of our new *RBF/GLM* surrogate *model.*

Training the RBF/GLM model starts with construction of two auxiliary models: the first, *global* RBF network $\hat{f}_{\mathrm{RBF}} : \mathbf{S}^{(\mathrm{C})} \to \mathbb{R}$ is fitted on the continuous input variables while the second, GLM $\hat{f}_{\mathrm{GLM}} : \mathbf{S}^{(\mathrm{D})} \to \mathbb{R}$ is built using the discrete variables. Both of them make use of all the available training data and regress the response-variable values.

Global RBF network. Training of the auxiliary RBF network works similarly to the training of the RBF networks in the previous RBF/DSCL model [1] – the same starting values for centers and weights, and cross-validation for choosing the best number of components g is used. However, instead of clusters, all the data in the database $\mathbf{D}$ are used at once.

GLM model. Generalized linear model is used in its continuous-response form and responses are supposed from normal distribution $\mathbf{Y} \sim N(\boldsymbol{\mu}, \sigma^2)$. Even though the latter assumption generally does not hold, GLMs still provide useful mean of regression expressed on the basis of the discrete values.

Before using or fitting the GLM, the discrete values must be converted to a proper representation. Since we do not expect any ordering on the discrete values, we have chosen *dummy coding* [13] which establishes one binary indicating variable $I_{ij} \in \{0,1\}$ for each nominal value from the value sets $V_s(X_i^{(\mathrm{D})})$, $i = 1, \dots, d$, $j = 1, \dots, |V_s(X_i^{(\mathrm{D})})|$ of the original discrete variables. Assignment between the original discrete values and the dummy coding

$$\mathit{dummy} : \mathbf{S}^{(\mathbf{D})} \to \{0,1\}^{|V_s(X_1)|+\ldots+|V_s(X_d)|} \qquad (4)$$

has to be recorded for evaluation with the surrogate model.

Final RBF clustered model. Having created the *global* RBF network $\hat{f}_{\mathrm{RBF}}$ and the GLM model $\hat{f}_{\mathrm{GLM}}$, we can proceed with the construction of the *final* RBF clustered surrogate model $\hat{f} : \mathbf{S}^{(\mathrm{C})} \to \mathbb{R}$. The process starts with clustering of the training data from the database $\mathbf{D} = \{\mathbf{x}_i^{(\mathrm{D})}, \mathbf{x}_i^{(\mathrm{C})}, y_i\}_{i=1}^{N}$ according to the difference between responses of the two auxiliary models on the corresponding input variables (for $i = 1, \dots, N$)

$$\mathit{diff}_i = \hat{f}_{\mathrm{RBF}}(\mathbf{x}_i^{(\mathrm{C})}) - \hat{f}_{\mathrm{GLM}}(\mathit{dummy}(\mathbf{x}_i^{(\mathrm{D})})). \qquad (5)$$

The sizes of the clusters have to be at least $s_{\min}$ – the minimal number of data needed for fitting one RBF network. This number is provided by the user and its best value depends on a particular task. The higher the $s_{\min}$ is, the more components can each RBF network have, but the more distinct discrete values are usually grouped together in one cluster.

```
FitTheModel($s_{\min}$, $\mathbf{D}$, $e$)
Arguments: $s_{\min}$ – min. size of clusters,
    $\mathbf{D}$ – database, $e$ – type of error estimate:
    MSE, AIC, or BIC
Steps of the procedure:
  (1) $(\hat{f}_{\mathrm{RBF}}, \mathit{rbf}_{\mathrm{GLOB}}) \leftarrow$ fit the global RBF
  (2) $(\hat{f}_{\mathrm{GLM}}, \mathit{glm}) \leftarrow$ fit the GLM
  (3) $\{\mathit{diff}_i\}_{i=1}^{N} \leftarrow$ differences $(\hat{f}_{\mathrm{RBF}} - \hat{f}_{\mathrm{GLM}})$ on $\mathbf{D}$
  (4) $\{C_j\}_{j=1}^{m} \leftarrow$ cluster $\mathbf{D}$ into clusters of size
        at least $s_{\min}$ according to $\{\mathit{diff}_i\}_{i=1}^{N}$
  (5) for each cluster $C_j$, $j = 1, \dots, m$
  (6)     for $g_j = 1, \dots, g_j^{\max}$
  (7)         $\mathit{mse}[j, g_j] \leftarrow$ average MSE$_{CV}$ from
                fitting RBF with $g_j$ components
  (8)     $g_j^{\star} \leftarrow$ the number of components
            of the best RBF
  (9)     $\mathit{rbf}_j \leftarrow$ retrained RBF network
            with $g_j^{\star}$ components
 (10)     $\mathit{mse}_j \leftarrow \mathit{mse}[j, g_j^{\star}]$
Output: $\{\mathit{rbf}_{\mathrm{GLOB}}, \mathit{glm}, (\mathit{rbf}_j, \mathit{mse}_j, \mathit{diff}_j)_{j=1}^{m}\}$
```

Fig. 1. Pseudo-code of the fitting procedure.

One separate RBF network rbf_j is trained on the data of each cluster C_j, $j = 1, \dots, m$. The maximal number of components of each network is upper-bounded by $g_j^{\max} = \lfloor (\frac{k-1}{k} |C_j|)/\rho \rfloor$. Training these networks is analogous to training of the *global* RBF network described in Section 3.1. The only difference is in the training data: only the data of individual clusters are used for each network.

3.2 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 $\{\mathit{rbf}_{\mathrm{GLOB}}, \mathit{glm}, (\mathit{rbf}_j, \mathit{mse}_j, \mathit{diff}_j)_{j=1}^{m}\}$. Here, $\mathit{rbf}_{\mathrm{GLOB}}$ are *global* RBF network parameters, $\mathit{glm} = (\beta_0, \dots, \beta_r)$ are parameters of the GLM, rbf_j global RBF network parameters, mse_j are the MSE$_{CV}$ obtained from cross-validation, and diff_j are the difference diff (5) averaged on the j-th cluster's data.

Given a new individual $(\tilde{\mathbf{x}}^{(\mathrm{C})}, \tilde{\mathbf{x}}^{(\mathrm{D})})$, evaluation with the surrogate model starts with computing the difference between responses of the *global* RBF network and GLM with corresponding parameters $\mathit{rbf}_{\mathrm{GLOB}}$ and glm

$$\begin{aligned}\widetilde{\mathit{diff}} = \ & \hat{f}_{\mathrm{RBF}}(\tilde{\mathbf{x}}^{(\mathrm{C})}; \mathit{rbf}_{\mathrm{GLOB}}) \\ & - \hat{f}_{\mathrm{GLM}}(\mathit{dummy}(\tilde{\mathbf{x}}^{(\mathrm{D})}); \mathit{glm}). \qquad (6)\end{aligned}$$

Based on this value, the index c of the cluster with the average difference most similar to the individual's difference is obtained

$$c = \arg \min_{j=1,\dots,m} |\mathit{diff}_j - \widetilde{\mathit{diff}}|. \qquad (7)$$

Finally, the response of the c-th *final* RBF network is used as a return value of the surrogate model

$$\tilde{y} = \hat{f}(\tilde{\mathbf{x}}^{(\mathrm{C})}; \mathit{rbf}_c) = \sum_{i=1}^{g_c^{\star}} \pi_{ic} f_{ic}(||\tilde{\mathbf{x}}^{(\mathrm{C})} - \mathbf{c}_{ic}||). \qquad (8)$$

If more than one cluster is at the same distance from the individual, the RBF network with the lowest MSE$_{CV}$ is chosen.

4 Implementation and results of testing

Our algorithms were implemented in the MATLAB environment. We have been utilizing the Global Optimization Toolbox which provided us with a platform for testing the model on a benchmark optimization task. Similarly, our hierarchical clustering method extends the cluster analysis from the Statistical Toolbox which 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 three different kinds of data. The first two datasets (Valero and HCN) are the same as in our last article [1], the third is the building1 dataset from Proben1 [18] collection.

Valero's [20] benchmark fitness function was constructed to resemble empirical fitness functions from chemical engineering. The surrogate models have been 10-times trained on dataset with 2000 randomly generated data. Using the same settings for fitting, the average root of the MSE (RMSE) of the new RBF/GLM model has been only slightly decreased. (see Table 1 and the top graphs on Fig. 2).

Valero	RBF/GLM	**14.046** ± 1.0435
	RBF/DSCL	14.499 ± 1.518
HCN	RBF/GLM	**10.340** ± 1.866
	RBF/DSCL	15.620 ± 1.519
building1	RBF/GLM	**0.06407** ± 0.00496
	RBF/DSCL	0.13618 ± 0.00455

Table 1. Surrogate-models' average regression RMSE on Valero's benchmark, HCN data and building1 dataset.

The second dataset is from a real application in chemical engineering (cf. using RBF networks in this application area e.g. in the work of Jun [11]): the optimization of chemical catalysts for Hydrocyanic acid (HCN) synthesis [14]. Solutions of this task are composed of two discrete and 11 continuous variables, the whole dataset has 696 items. Fitting results are substantially different from the benchmark problem (considering the average response in the dataset $\bar{y} = 31.17$, the measured RMSE's are relatively much higher: see middle row of graphs on Fig. 2). RMSE of the new RBF/GLM model has been decreased by nearly 35 % comparing to the previous model's error.

Prechelt's Proben1 [18] is a popular collection of datasets for data mining, originally intended for neural networks benchmarking. We have tested our models on the building1 dataset using the first response variable indicating electrical energy consumption in a building; multiple-trained on the first 3156 and tested on the remaining 1052 data, as suggested by Prechelt. Average results from ten trainings show that the former RBF/DSCL model is not able to sufficiently express the relation between discrete variables and the output. Conversely, the results of the RBF/GLM model are at least comparable to the results reported elsewhere [12, 2, 15].

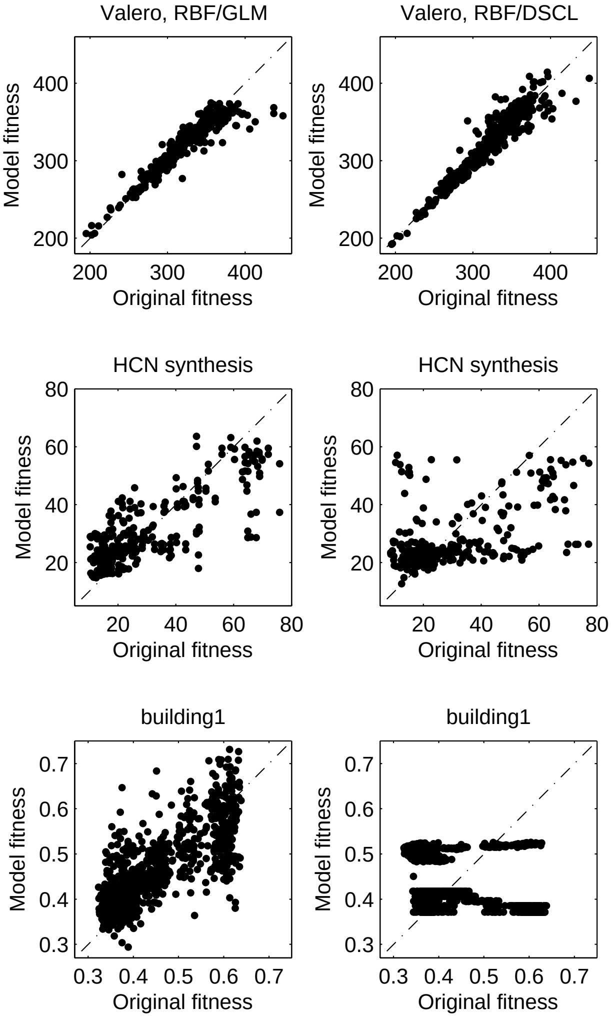

Fig. 2. Scatter plots of the RBF/GLM (left column) and RBF/DSCL model (right column) on testing data.

4.2 Genetic algorithm performance on the benchmark fitness

The benchmark fitness enabled us to test the model with the GA [1]. As shown in Table 2, the GA with the surrogate model reaches on this function the same fitness values as the non-surrogate GA using only less than 30 per cent of the original fitness function evaluations (generation-based EC), or it is able to find 1.1-times better solution with 80 per cent of the original fitness evaluations (individual-based EC).

EC settings of the surrogate model	fitness of the best found individual	number of orig. fitness evaluations
without model	486.38 ± 56.5	4130 ± 1546
individual-based	544.73 ± 3.9	3241 ± 926
generation-based	490.28 ± 44.9	1185 ± 358

Table 2. GA performance without surrogate model and with the RBF/DSCL-based model; average results from 100 runs of the algorithm

5 Conclusion

Two kinds of surrogate models of expensive objective functions for mixed-variable continuous and discrete optimization were presented in this paper. Both of them make use of RBF networks; the first model focuses training of the RBF networks using clustering on the discrete part of the data while the second builds GLM on the discrete input variables. Detailed algorithms for training the models were provided. Results of testing on three different datasets showed that especially the second model is a competitive kind of regression for costly objective functions. Using the model on the benchmark fitness function resulted in saving up to 70 per cent of the original evaluations or 10 per cent increase of the final solution quality.

One of the most similar works dealing with surrogate models is the paper of Zhou [22]. He uses RBF networks as a local surrogate model in combination with a global model based on Gaussian processes. Other literature employs polynomials [8], Gaussian processes [4], or multilayer perceptron networks [10], but most publications consider only continuous or continuous and integer optimization.

References

1. L. Bajer, M. Holeňa: *Surrogate model for continuous and discrete genetic optimization based on RBF networks.* In C. Fyfe et al.,(Ed.), Intelligent Data Engineering and Automated Learning – IDEAL 2010, vol. 6283 of Lecture Notes in Computer Science, pp. 251–258. Springer, September 2010.
2. J.L. Bernier, J. Ortega, I. Rojas, E. Ros, A. Prieto: *Obtaining fault tolerant multilayer perceptrons using an explicit regularization.* Neural Processing Letters, 12(2), October 2000, 107–113.
3. A. J. Booker, J. Dennis, P.D. Frank, D. B. Serafini, V. Torczon, M. Trosset: *A rigorous framework for optimization by surrogates.* Structural and Multidisciplinary Optimization 17, 1999, 1–13.
4. D. Buche, N.N Schraudolph, P. Koumoutsakos: *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.
5. M.D. Buhmann: *Radial basis functions: theory and implementations.* Cambridge Univ. Press, 2003.
6. D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq: *A surrogate modeling and adaptive sampling toolbox for computer based design.* Journal of Machine Learning Research 11, 2010, 2051–2055.
7. M. Holeňa, T. Cukic, U. Rodemerck, D. Linke: *Optimization of catalysts using specific, description-based genetic algorithms.* Journal of Chemical Information and Modeling 48(2), 2008, 274–282.
8. S. Hosder, L.T. Watson, B. Grossman: *Polynomial response surface approximations for the multidisciplinary design optimization of a high speed civil transport.* Optimization and Engineering 2(4), 2001, 431–452.
9. Y. Jin: *A comprehensive survey of fitness approximation in evolutionary computation.* Soft Computing – A Fusion of Foundations, Methodologies and Applications 9(1), 2005, 3–12.
10. Y. Jin, M. Hüsken, M. Olhofer, B. Sendhoff: *Neural networks for fitness approximation in evolutionary optimization.* Knowledge Incorporation in Evolutionary Computation, 2005, 281.
11. Q. Jun-Fei, Han Hong-Gui: *A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modeling.* International Journal of Neural Systems 20(1), 2010, 63–74.
12. Cheol-Taek Kim, Ju-Jang Lee: *Training two-layered feedforward networks with variable projection method.* IEEE Transactions on Neural Networks 19(2), 2008, 371–375.
13. P. McCullagh, J.A. Nelder: *Generalized linear models.* Chapman & Hall, 1989.
14. S. Möhmel, N. Steinfeldt, S. Endgelschalt, M. Holeňa, S. Kolf, U. Dingerdissen, D. Wolf, R. Weber, M. Bewersdorf: *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.
15. P. Narasimha, W.H. Delashmit, M.T. Manry, Jiang Li, F. Maldonado: *An integrated growing-pruning method for feedforward network training.* Neurocomputing 71(13-15), August 2008, 2831–2847.
16. M. Olhofer, T. Arima, T.S. B. Sendhoff, G. Japan: *Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies.* Evolutionary Design and Manufacture, 45–54, 2000.
17. Y.S. Ong, P.B. Nair, A.J. Keane, K.W. Wong: *Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems.* Knowledge Incorporation in Evolutionary Computation, 307–332, 2004.

18. L. Prechelt: *Proben1: A set of neural network benchmark problems and benchmarking rules.* Technical Report 21/94, 1994.
19. G. Singh, R. V. Grandhi: *Mixed-variable* optimization strategy employing multifidelity simulation and surrogate models. AIAA Journal 48(1), 2010, 215–223.
20. S. Valero, E. Argente, V. Botti: *DoE framework for catalyst development based on soft computing techniques.* Computers & Chem. Engineer. 33(1), 2009, 225–238.
21. A. Younis, Z. M. Dong: *Global optimization using mixed surrogate models for computation intensive designs.* In 2nd International Symposium on Computational Mechanics Hong Kong 1233, 2010, 1600–1605.
22. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane: *Combining global and local surrogate models to accelerate evolutionary optimization.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37(1), 2007, 66–76.