BUILDING ENSEMBLES OF SURROGATES BY OPTIMAL CONVEX COMBINATION

Martina Friese and Thomas Bartz-Beielstein
SPOTSeven Lab, TH Köln
Steinmüllerallee 1, 51643 Gummersbach, Germany
{martina.friese|thomas.bartz-beielstein}@th-koeln.de

Michael Emmerich
LIACS, Leiden University
Niels Bohrweg 1, 2333CA Leiden, The Netherlands
m.t.m.emmerich@liacs.leidenuniv.nl

Abstract When using machine learning techniques for learning a function approximation from given data it can be difficult to select the right modelling technique. Without preliminary knowledge about the function it might be beneficial if the algorithm could learn all models by itself and select the model that suits best to the problem, an approach known as automated model selection. We propose a generalization of this approach that also allows to combine the predictions of several surrogate models into one more accurate ensemble surrogate model. This approach is studied in a fundamental way, by first evaluating minimalistic ensembles of only two surrogate models in detail and then proceeding to ensembles with more surrogate models. The results show to what extent combinations of models can perform better than single surrogate models and provide insights into the scalability and robustness of the approach. The focus is on multi-modal functions which are important in surrogate-assisted global optimization.

Keywords: Function Approximation, Surrogate Models, Model Selection, Ensemble Methods, Global Optimization

1. Introduction

Surrogate models are mathematical functions that, basing on a sample of known objective function values, approximate the objective function, while being cheaper in terms of evaluation. Such surrogate models can then be used to partially replace expensive objective function evaluations. Expert systems like SPOT [1] come with a large variety of models that has to be chosen from when initiating an optimization process. The choice of the right model implies the quality of the the optimization process.

Often expert knowledge is needed to decide which model to select for a given problem. If there is no preliminary knowledge about the objective function it might be beneficial if the algorithm could learn all by itself which model suits best to the problem. This can be done by evaluating different models on test data a priori and using a statistical model selection approach to select the most promising model.

Some occurrences imply that there might also be a benefit in linearly combining predictors from several models into a more accurate predictor. In Figure 1 such an occurrence is happening. Predictions with two different (Kriging) models are shown and results obtained by a convex combination of the predictors of these models. Different errors seem to be compensated by the combined model 's predictions.

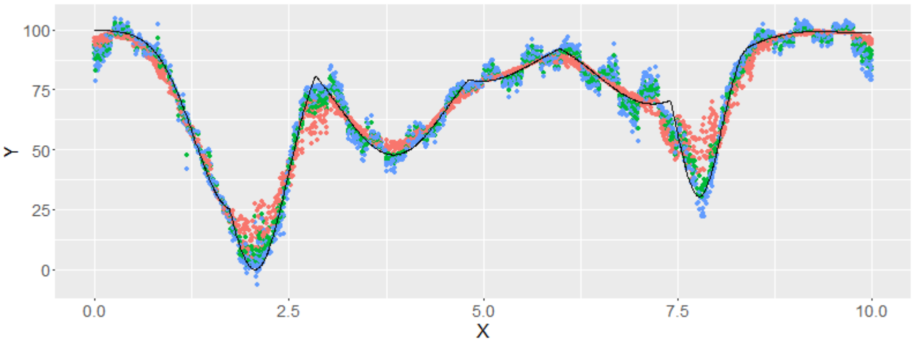


Figure 1: *The black line marks the actual objective function value. The dots show the results obtained in a leave-one-out cross-validation. Blue and red dots mark the predictions of single models. The green dots shows predictions obtained with an optimal convex linear combination of the two predictors.*

Such occurences show that a predictor based on a single modeling approach is not always the best choice. On the other hand, complicated expressions based on multiple predictors might not be a good choice, either, due to overfitting and lack of transparency. Using convex combinations of predictors from available models seems to be a 'smart' compromise. Given s surrogate models $\hat{y}_i : \mathbb{R}^d \to \mathbb{R}, i = 1, \ldots, s$ and d the dimension of the search space, by a *convex combination of models* we understand a model given by $\sum_{i=1}^{s} \alpha_i \hat{y}_i$ with $\sum \alpha_i = 1$ and $\alpha_i \geq 0, i = 1, \ldots, s$. Finding an optimal convex combination of models can be viewed as a generalization of model selection, where selecting only one model is a special case. Convex combinations of predictors have also the advantage that they combine only predictions and can be used for heterogeneous model ensembles. The main research questions are:

(Q-1) Can convex combinations of predictors improve as compared to (single) model selection?

(Q-2) Given the answer is positive, what are explanations of the observed behavior?

(Q-3) How can a system be build that finds the optimal convex combination of predictions on training data?

In order to answer these questions, detailed empirical studies are conducted, starting from simple examples and advancing to more complex ones. This paper follows a structure, where the discussion of experimental results follows directly the introduction of the modeling extensions.

2. General Approach and Related Work

To base a decision or build a prediction from multiple opinions is common practice in our everyday live. It happens in a democratic government, or when in TV shows the audience is asked for help. One also might use it when we try to build an opinion on a topic that is new to us. Naturally, such tools already found their way into statistical prediction and machine learning. In statistics and machine learning an *ensemble* is a prediction model from several models, aiming for better accuracy. A comprehensive introduction to ensemble-based approaches in decision making is given in [9] and [5]. Generally, there are two groups of ensemble approaches: the first group's approaches, the so-called *single-evaluation* approaches, only choose and build one single model, whereas the second group's approaches, the so-called *multi-evaluation* approaches, build all models, and use the derived information to decide which output to use. For each of these two approaches, several model selection strategies can be implemented. Well-known strategies are:

- *Round robin* and *randomized choosing* are the most simplistic implementations of ensemble-based strategies. In the former approach, the models are chosen in a circular order independent of their previously achieved gain. In the latter approach, the model to be used in each step is selected randomly from the list of available models. The previous success of the model is not a decision factor.
- *Greedy strategies* choose the model that provided the best function value so far, while the SoftMax strategy uses a probability vector, where each element represents the probability for a corresponding model to be chosen [13]. The probability vector is updated depending on the reward received for the chosen models.

- *Ranking strategies* try to combine the responses of all meta models to one response, where all meta models contributed to, rather than to choose one response.
- *Bagging* combines results from randomly generated training sets and can also be used in function approximation, whereas
- *Boosting* combines several weak learners to a strong one in a stochastic setting.
- *Weighted averaging* approaches do not choose a specific model's result but rather combine it by averaging. Since bad models should not deteriorate the overall result, a weighting scheme is introduced. Every model's result for a single design point is weighted by its overall error, the sum over all models yields the final value assigned to the design point. A similar approach is *stacking*, where the weights are chosen by an additional training step.

The convex model combinations in this paper can be viewed as an elegant stacking approach and as such is similar to 'ensembles of surrogates' [7], which however used a fixed rule for determining weights. In our work weights are optimized globally and the approach is analysed in a controlled and detailed way. Since most of the black-box real-world problems considered to be difficult are multimodal, the focus for this work also is on multimodal function approximation (cf. [12, 14, 10, 8]).

3. Preliminaries

By a *surrogate model*, we understand here a function $\hat{y} : \mathbb{R}^d \to \mathbb{R}$ that is an approximation to the objective function $y : \mathbb{R}^d \to \mathbb{R}$, learned from a finite set of evaluations of the objective function. Kriging surrogate models are used in our study. A set of three different kernels is used to implement the ensemble strategies. Following the definitions from [11], the correlation models can be described as follows. We consider stationary correlations of the form $\mathcal{R}(\theta, w, x) = \prod_{j=1}^{n} \mathcal{R}(\theta_j, w_j - x_j)$. The first model uses the *exponential* kernel $\mathcal{R}(\theta, w, x) = \exp(-\theta_j|w_j - x_j|)$ the second model uses an *gaussian* kernel $\mathcal{R}(\theta, w, x) = \exp(-\theta_j|w_j - x_j|^2)$, whereas the third model is based on the *spline correlation* function $\mathcal{R}(\theta, w, x) = \zeta(\theta_j|w_j - x_j])$ with

$$\zeta(\epsilon_j) = \begin{cases} 1 - 15\epsilon_j^2 + 30\epsilon_j^3 & \text{for} \quad 0 \leq \epsilon_j \leq 0.2 \\ 1.25(1 - \epsilon_j)^3 & \text{for} \quad 0.2 < \epsilon_j < 1 \\ 0 & \text{for} \quad \epsilon_j \geq 1. \end{cases}$$

Here, ϵ and θ are hyperparameters estimated by likelihood maximization.

For generating *test functions* we use the *Max-Set of Gaussian Landscape Generator* (MSG). It computes the upper envelope of m weighted

Table 1: *Gaussian landscape generator options*

Parameter	*Description*	*Value*
d	Dimension	$2-10$
m	Number of peaks	$10-40$
l	Lower bounds of the region, where peaks are generated	$\{0_1,\ldots,0_d\}$
u	Upper bounds of the region, where peaks are generated	$\{5_1,\ldots,5_d\}$
max	Max function value	100
t	Ratio between global and local optima	0.8

Gaussian process realizations and can be used to generate continuous, bound-constrained optimization problems [6].

$$G(x) = \max_{i \in 1,2,\ldots,m}(w_i g_i(x)),$$

where $g : \mathbb{R}^n \to \mathbb{R}$ denotes an n-dimensional Gaussian function

$$g(x) = \left(\frac{\exp\left(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right)}{(2\pi)^{n/2}|\Sigma|^{1/2}} \right)^{1/n},$$

μ is an n-dimensional vector of means, and Σ is an $(n \times n)$ covariance matrix. Implementation details are presented in [2]. For the generation of the objective function the spotGlgCreate method of the SPOT package has been used. The options used for our experiments are shown in Table 1. With the parameter d the dimension of the objective function is specified. The lower and upper bounds (l and u, respectively) specify the region where the peaks are generated. The value max specifies the function value of the global optimum, while the maximum function value of all other peaks is limited by t, the ratio between the global and the local optima.

4. Binary Ensembles

This Section analyses models which combine only two models. Convex combinations of models will be referred to as ensemble models, while the original models will be referred to as base models. We focus on positive weights, since we do not want to select models that make predictions which are anti-correlated with the results.

A sample of points (design) is evaluated on the objective function (MSG, for parameters see Table 1). For the sampling of the points a latin hypercube design featuring 40 design points is generated. The two base models are Kriging with exponential correlation function (referred to as a) and gaussian correlation function (referred to as b). Both base models are fitted to the data and then asked to do a prediction on the testdata. The predictions $\hat{y}$ of the ensemble models are calculated as convex combinations of the predictions of the base models.

Given a weight α_i, where $\alpha_i \in \{0.0, 0.1, 0.2, ..., 0.9, 1.0\}$, the ensemble models can be defined as the linear combinations of the models a and b as follows:

$$\hat{y}_n = \alpha_n \times \hat{y}_a + (1 - \alpha_n) \times \hat{y}_b \tag{1}$$

The models are evaluated by calculating the root mean squared error (RMSE) of the predictions made during a leave-one-out cross-validation on the 40 design points.

Since randomness has been induced into the experiment by using the latin hypercube design, the evaluation process has been repeated 50 times. With each model returning one prediction for each design point in every repetition this results in a total of 2000 prediction values (40 design points × 50 repetitions) for each model.

To get a first quick insight into the result data, for each repetition the rankings of the RMSE's have been calculated. The models with $\alpha = 0.6$, $\alpha = 0.8$ and $\alpha = 0.9$ dominate this comparison, each performing best 8 out of 50 times. The base models, a and b, performed best only in four respectively two cases out of 50. Never an ensemble model with positive weights was performing worst.

In order to achieve some comparability between the RMSE's of different repetitions all RMSE's have been repetition-wise scaled to values between zero and one, so that the scaled RMSE of the best model in one repetition is always zero and the scaled RMSE of the worst model for one repetition is always 1.0. Figure 2 shows the boxplot over these

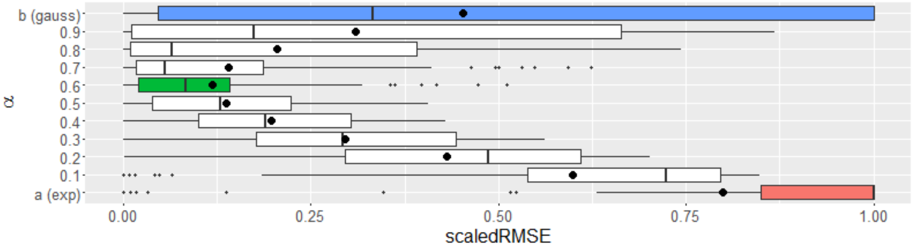


Figure 2: *Boxplot over the scaled RMSE's of all models. The models are defined by an α-weighted linear combination of the two base models. The results of the base models depicted on the outer rows and colored red (exponential kernel), respectively blue (gaussian kernel). The model combination chosen as best with $\alpha = 0.6$ is colored green. The mean value of each result bar is marked by a dot.*

scaled RMSE's. It can be seen that the model a (exponential) in most of the cases performs worst since its median is 1.0 - only some outliers come closer to zero.

Model b (gaussian) shows a larger variation in its performance. It has been the best- as well as the worst performing model each at least once. Its median and mean performances are average in comparison with all models evaluated. A parabolic tendency can be seen in the performance. Due to the convex combination of the predictor, a prediction by the ensemble model cannot be worse but it might be better than both base models. An ensemble can only be better, if one model overestimates and the other model underestimates the objective function value. In the experiment this happens in 649 out of 2000 cases.

As a *consistent method for evaluating the performance and automatically choosing the best model* the following approach is proposed: Model-wise mean-, median- and 3rd quartile-values are calculated. The resulting values are ranked and the rankings summed up to one final ranking. The model that achieved the lowest value is recommended as best choice. In Figure 2 the model recommended as best choice by this method is colored green.

5. Detailed Analysis on Transparent Test Cases

It can clearly be stated that for this first experiment setup the combination of two models is beneficial for the overall prediction. In this section we're going to have a closer look at possible explanations for the successful result. Are there problem features that encourage using ensembles and is this result generalizable.

We chose a 1D objective function to allow for a better understanding of the underlying process. This is the only change in the experimental setup. The Figures 1 and 2 from Section 4 depict the main results of this second experiment setup. Figure 2 shows the scaled RMSE's for all models. Applying the rule defined in Section 4 names the model obtained by a linear combination with $\alpha = 0.7$ as best choice.

Figure 1 shows only the performance of the best choice model and the base models. Each dot marks a single prediction made during the leave-one-out cross-validation. As can be seen in the plot, the predictions of the model a (exponential), marked by red dots, seem to smooth the objective function - straight segments are well met while curved segments are smoothed out.

The predictions of the model b (gaussian), marked by the blue dots show signs of overfitting. Again straight segments are well met but when approaching local extrema the predictions start to oscillate. So the linear combination of both predictions averages positive as well as negative outliers of base models. This seems to provide some benefit to the overall experiment outcome.

Since the curves and corners in the objective function seem to make the game here, two additional experiments are set up. For these experiments two objective functions are specified featuring corners that are not continuous differentiable. For one experiment a triangle objective function is used while the other features a piecewise assembled objective function. Figure 3 shows the results for the piecewise assembled objective function. Looking at these results, we again find a strong parabolic tendency in the boxplot. Both base models have a rather large variance in their performance. The ensemble model marked as best choice has a smaller variance and performed better than the base models in nearly all cases. The results on the triangle objective function happened to show a clear tendency towards base model b, which clearly outperformed basemodel a and thus was chosen best.

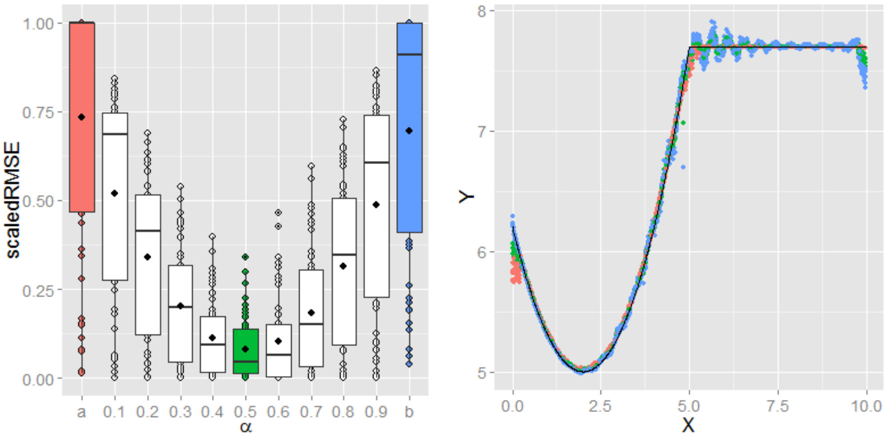


Figure 3: *Results on a piecewise assembled objective function. Left hand side plot shows the scaled RMSE's. The α value defines the weight for the linear combination. The ensemble obtained by a linear combination with $\alpha = 0.5$, here colored green, is suggested best for this experiment setup. On the right hand side all predictions done during the leave-one-out cross validation for the base models and the best model are plotted against the objective function.*

6. Ternary Ensembles

Next, the experiments are extended to a larger scale: The dimensionality of the objective function is increased and three base models are combined. As before Kriging models with different kernels are used, but now a third model using the spline correlation function is added.

$$\alpha_n, \beta_n, \gamma_n \in \{0.0, 0.1, 0.2, ..., 0.9, 1.0\}, \quad \alpha_n + \beta_n + \gamma_n = 1 \tag{2}$$

For the linear combination of three base models three weights are needed, that sum up to one as specified in (2). With a step size of 0.1 for the linear combinations this results in 66 models.

Figure 4a shows the results of the first experiment using three base models. The only change that has been made to the original experiment setup, besides the number of base models, is the dimension d of the objective function and the number of peaks m generated in the gaussian landscape. As a first step towards objective functions of higher complexity, the dimension of the objective function has been set to 4. But this change alone is not sufficient to gain a larger complexity, since without adjusting the number of gaussian components used for generating the objective function, it rather gets less complex. Thus the number of gaussians process trajectories is adjusted to ten times the dimension.

With the points getting smaller when approaching the center of the triangle, it can be stated, that again it is beneficial to use a convex combination of the base models.

7. Scaling-up to multiple models

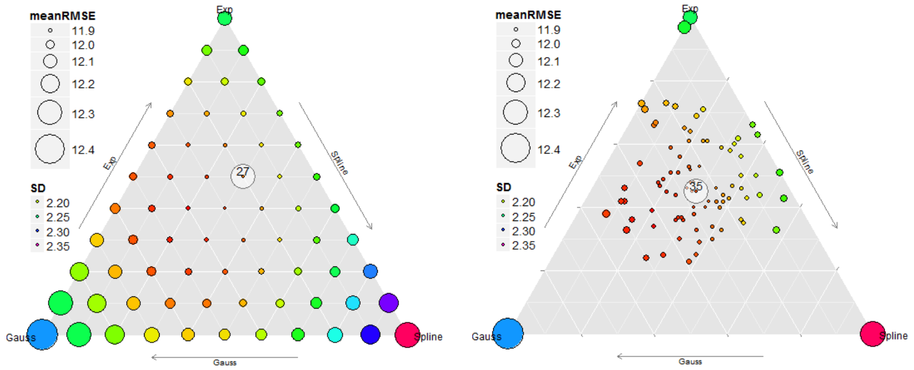
By now, only experiments with up to three models are carried out, but the underlying goal is to evolve a system that is able to handle quite a large set of available base models. But at this point quickly another approach is needed, since the number of possible discretised convex combinations between a higher number of base models grows exponentially. A recursive formula is given below: There is only one setting where the first model gets all the weight (first factor in sum). In all other settings the remaining weight must be distributed on the remaining models.

$$f(r, s) = 1 + \sum_{r^*=1}^{r-1} f(r - r^*, s - 1), \quad f(r, 1) = 1, f(1, s) = s \tag{3}$$

The relation between number of models, the step size for the discretised convex combinations and the resulting number of linear combinations can be expressed as function of r the reciprocal of the step size and s the number of models as defined in (3). Using three base models and a step size of 0.1 as defined in (2) this results in $f(10, 3) = 66$ linear combinations. Now thinking of combinations of 10 base models already results in $f(10, 10) = 92378$ linear combinations.

The complexity of the search space, when increasing the number of models, quickly gets too large to do a complete evaluation of all possible

convex combinations with a fixed step size of 0.1. Looking at previous results, the function that describes the performance of the models built by convex combinations up to this point only showed unimodal characteristics. This seems to be expectable due to the nature of convex combinations. We expect the function to show this characteristic also when combining larger number of models.
Thus, instead of a complete evaluation of all linear combinations, an optimization step is implemented to find the best combination. The allowed weights are restricted to a precision of two decimal places. Since the area around the optimum tends to build a plateau. This reduces the possible search space without loosing the possible best solution.



(a) *The optimal linear combination has been found by a complete evaluations of all linear combinations using a fixed step size of 0.1.*

(b) *The plot shows the results of the same experiment setup as presented in Section 6. The optimal linear combination has been searched with a simple (1+1)-Evolution Strategy with 1/5th success rule (cf. [3]).*

Figure 4: *The plots show the results of the experiment set up with three base models. Each circle depicts the performance results for one model. The three base models are located on the corners of the triangle, models gained by linear combinations of only two models are located on the outer border. Circles on the inner area of the area show the results for models that were gained by linear combinations of all three base models. The size of the circles denotes the mean RMSE value, the color the standard deviation. The model proposed as best choice is marked by an additional white circle.*

For the sake of comparability, the experiment setup here is exactly the same as the one used in Section 6. Only the process itself changed. Prior to this experiment, all convex combinations have been evaluated. Now, only the base models are evaluated initially. Other models are only evaluated during the optimization. We also stuck to the method used

by the (1+1)-ES of comparing the offspring only to the parent rather than to the whole population as we did it before.
For the mutation of the weights vector $\vec{v} = (\alpha, \beta, \gamma)^T$ three random samples of a normal distribution function with standard deviation of 0.16 have been drawn and added to the weight vector. Since this alone does not meet the requirements needed for a valid weight vector, the resulting vector has been adjusted in three steps:

1) If $\min(\alpha, \beta, \gamma) < 0$ then $\vec{v} \leftarrow \vec{v} - (\min(\alpha, \beta, \gamma), \ldots, \min(\alpha, \beta, \gamma))^T$,
2) $\vec{v} \leftarrow \vec{v}/(\alpha + \beta + \gamma)$,
3) Round the values α, β, γ to two decimal places so, that $\alpha + \beta + \gamma = 1$.

For this experiment we allowed a maximum of 100 individuals to be evaluated. Within these bounds already the 35th evaluated individual has been the best individual found in this run. Figure 4b depicts the results of this optimization step. As before, the best individual is marked by a white circle. However, since determination of optimal weights in the linear model is a non-linear optimization problem, we cannot guarantee the optimality of the proposed weights. So far, we have achieved similar results in repeated runs and on different objective functions. Due to space constraints, statistical validation is however left to future work.

8. Discussion and Outlook

Reconsidering the research questions from Section 1, it was shown that convex linear combinations of predictors can generate better results than model selection (Q-1). A system, which finds optimal linear combinations, was presented in Section 4. As a possible explanation a compensation of outliers was found, an effect that occured in particular in non-smooth objective functions (Q-2). The corresponding experiments were extended to a larger scale, in terms of dimensionality as well as number of models, in Section 6 with results indicating that the methods are scalable (Q-3). Finally, in Section 7, we proposed a method to include even more base models to the system, showing that evolutionary optimization can be an effective tool for finding optimal convex combinations. With this method the foundation has been created for a larger system including all available models. Although research questions (Q-2) and (Q-3) could be partially answered, larger studies are required to statistically confirm scalability and find in depth explanations.
In summary, convex combination of models are a promising approach in situtations where several types of models are available. if the user does not know, which model to choose, a linear combination might be a promising approach. An interesting aspect about convex combinations is that they are easy to interpret and that weights in the linear model

can shed some light on the relevance of certain models and illustrate, which model is active.

Ideas and questions that will be discussed in future work are:

- Experiments featuring more base models, also including other types of models.
- Extensive analysis of the influence of objective function attributes on the experiment outcome. The results of Section 5 suggest, that particularly piecewise assembled objective functions might be of special interest.
- Studies also allowing other operations than simple convex combinations only: Does increasing the model complexity of model combinations yield much better results?
- Comparing to approaches that chose fixed weights [7].

Acknowledgement This work has been supported by the Bundesministeriums für Wirtschaft und Energie under the grants KF3145101WM3 und KF3145103WM4. This work is part of a project that has received funding from the *European Union's Horizon 2020 research and innovation program* under grant agreement No 692286.

References

[1] T. Bartz-Beielstein. Spot: An r package for automatic and interactive tuning of optimization algorithms by sequential parameter optimization. Technical Report 05/10, Research Center CIOP (Computational Intelligence, Optimization andData Mining), Cologne University of Applied Science, Faculty of Computer Scienceand Engineering Science, 2010. Comments: Related software can be downloaded from http://cran.r-project.org/web/packages/SPOT/index.html.

[2] T. Bartz-Beielstein. How to Create Generalizable Results. In J. Kacprzyk and W. Pedrycz, editors, Springer Handbook of Computational Intelligence, pages 1127–1142. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[3] H. G. Beyer and H. P. Schwefel (2002). Evolution strategies–A comprehensive introduction. Natural computing, 1(1), 3-52.

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

[5] M. Friese, M. Zaefferer, T. Bartz-Beielstein, O. Flasch, P. Koch, W. Konen, and B. Naujoks. Ensemble-Based Optimization and Tuning Algorithms. In F. Hoffmann and E. Hüllermeier, editors, Proceedings 21. Workshop Computational Intelligence, pages 119–134. Universitätsverlag Karlsruhe, 2011.

[6] M. Gallagher and B. Yuan. A general-purpose tunable landscape generator. IEEE Trans. Evolutionary Computation, 10(5):590–603, 2006.

[7] Goel, T., Haftka, R. T., Shyy, W., and Queipo, N. V. (2007). Ensemble of surrogates. Structural and Multidisciplinary Optimization, 33(3), 199-216.

[8] W. Jakob, M. Gorges-Schleuter, I. Sieber, W. Süß, H. Eggert. Solving a Highly Multimodal Design Optimization Problem Using the Extended Genetic Algorithm GLEAM. In: S. Hernandez, A. J. Kassab, C. A. Brebbia: Computer Aided Design

of Structures VI, WIT Press, Southhampton, Conf. Proc OPTI 99, S.205-214, 1999.

[9] R. Polikar. Ensemble based systems in decision making. Circuits and Systems Magazine, IEEE, 6(3):21–45, 2006.

[10] Ling Qing, Wu Gang, Yang Zaiyue, and Wang Qiuping. Crowding clustering genetic algorithm for multimodal function optimization. Applied Soft Computing, 8(1):88 – 95, 2008.

[11] J. S. Søren N. Lophaven, Hans Bruun Nielsen. Dace - a matlab kriging toolbox. Technical report, Technical University of Denmark, 2002.

[12] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu. Multimodal optimization by means of a topological species conservation algorithm. IEEE Transactions on Evolutionary Computation, 14(6):842–864, Dec 2010.

[13] R. S. Sutton and A. G. Barto. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[14] Ka-Chun Wong, Kwong-Sak Leung, and Man-Hon Wong. Protein structure prediction on a lattice model via multimodal optimization techniques. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, pages 155–162, New York, NY, USA, 2010. ACM.