Fuzzy Local Linear Approximation-based Sequential Design

Joachim van der Herten, Dirk Deschrijver and Tom Dhaene
Internet Based Communication Networks and Services (IBCN)
Ghent University - iMinds
Gaston Crommenlaan 8 (Bus 201), B-9050 Gent, Belgium
Email: joachim.vanderherten@intec.ugent.be

***Abstract*—When approximating complex high-fidelity black box simulators with surrogate models, the experimental design is often created sequentially. LOLA-Voronoi, a powerful state of the art method for sequential design combines an Exploitation and Exploration algorithm and adapts the sampling distribution to provide extra samples in non-linear regions. The LOLA algorithm estimates gradients to identify interesting regions, but has a bad complexity which results in long computation time when simulators are high-dimensional. In this paper, a new gradient estimation approach for the LOLA algorithm is proposed based on Fuzzy Logic. Experiments show the new method is a lot faster and results in experimental designs of comparable quality.**

I. Introduction

Simulating complex systems with multiple input and output parameters can be a very time-consuming process. For example, Ford Motor Company reported on a crash simulation for a full passenger car that takes 36 to 160 hours to compute [1]. Usually, little or no additional information is available on the inner working of the system. Use of such simulations for optimization, sensitivity analysis and understanding of the system is impractical.

To alleviate these limitations, the simulator approximating the complex system can be approximated with a surrogate model (or metamodel). This model mimics the original system but can be evaluated much faster. Construction of the model is performed by evaluating the simulator at key points in the input domain. The responses are used to fit one or more surrogate models: popular choices include polynomial or rational functions [2], Kriging Models [3], Artificial Neural Networks [4] and Radial Basis Functions [5]. The constructed surrogate model can now replace the simulator as it approximates the behaviour of the simulator over the entire domain. This approach is referred to as *global* surrogate modelling. In *local* surrogate modeling, local models are used to guide the optimization algorithm towards a global optimum. The local models are discarded afterwards.

The choice of the points (referred to as *samples*) in the input domain has a large impact on the quality of the resulting surrogate model. Ideally, the distribution maximizes the amount of information on the behaviour of the system. The configuration of the sample locations is called an experimental design. In traditional design of experiments the experimental design is generated prior to the modelling process. This is referred to as a *one-shot* design. After evaluation the model is constructed and the process ends. This approach holds a fundamental risk: if the model does not meet the required accuracy, the process has to be repeated from scratch. On the other hand, since nothing is known about the system we may evaluate more samples than required. Since each evaluation is expensive in terms of computation time, this is undesirable.

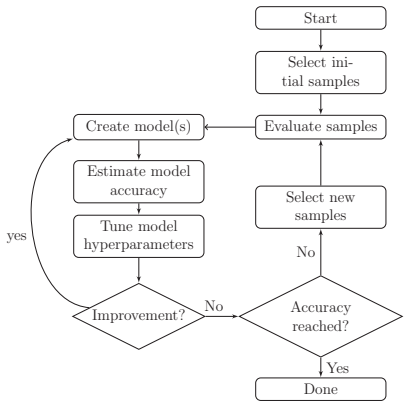


Fig. 1. Flow-chart of the SUMO Toolbox work-flow

A more popular choice is *sequential design* (or adaptive sampling). When the process starts, an initial design is evaluated and a surrogate model is constructed. When the surrogate model is not accurate enough new samples are chosen based on information acquired from prior evaluations and the intermediate model. Sequential design avoids the risk of under- or oversampling, and uses knowledge about the system as it becomes available during the process of determining where to sample next. This process is illustrated in Fig. 1. Two types of sequential design methods exist: *exploitation* -based methods focus on sampling regions in the input space that require special attention such as steep ridges and asymptotic or non-linear behaviour. *Exploration*-based methods on the other hand sample in regions that have been sampled sparsely. In practise, both concepts are important to obtain an accurate model. The LOLA-Voronoi method [6], [7] is a successful method for sequential design, combining an exploitation- and an exploration-based algorithm (LOLA and Voronoi respectively). The method is robust and generates designs with focus towards non-linear regions by estimating a gradient in each available point (LOLA). This methodology has proven to be successful in various research disciplines such as electromagnetic compatibility [8], [9], macromodeling of microwave systems [10], [11] and exposure assessment [12]. A drawback is high computational complexity, leading to a time-consuming sample selection process when many samples

978-1-4799-4509-2/14/$31.00 ©2014 IEEE

have already been evaluated, or the problem becomes high-dimensional.

This paper presents a modification to the LOLA-Voronoi algorithm which reduces computational demands. The resulting algorithm has the same desirable properties (in terms of required number of samples to obtain a certain model accuracy) but is applicable to problems of higher-dimensionality without causing long computation times. In the second section, the role of the gradient estimation in the LOLA algorithm is explained, and the computationally intensive parts are identified. Section three introduces a fast, new approach to estimate the gradient based on Fuzzy Logic. This approach is then integrated into the LOLA-Voronoi algorithm to result in a faster algorithm. This is illustrated by numerical results in Section 4.

II. Local linear approximation

The motivation behind the LOLA component of LOLA-Voronoi is the idea that regions with a dynamic response should be sampled more densely compared to very smooth regions, as illustrated in Fig. 2. The linearity is estimated by constructing a linear approximation in each sample and comparing this linear fit with nearby samples. If the fit is bad, this means the region surrounding the sample has a dynamic response and should be exploited. The LOLA algorithm does not incorporate the size of this region: if it is to big compared to regions surrounding other samples, the Voronoi part detects this and adds more samples in this region, which potentially leads to discovery of other non-linear regions, that can be exploited in the next iteration etc.

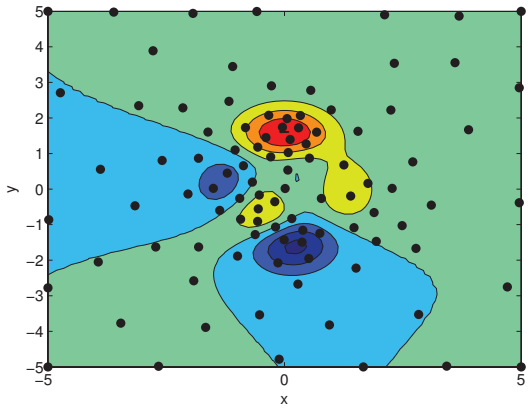


Fig. 2. Sequential design generated by LOLA-Voronoi for the Peaks function. Starting from a 12 point initial latin hypercube, samples were added one at a time. The sample density in the central non-linear region is higher compared to the flat surrounding.

A function $f : \mathbb{R}^d \to \mathbb{C}$, representing the complex system, is sampled at a set $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\}$ of N points. The best local linear approximation is the gradient defined as

$$\nabla f = \left(\frac{\partial f}{\partial x^{(1)}}, \frac{\partial f}{\partial x^{(2)}}, ..., \frac{\partial f}{\partial x^{(d)}} \right).$$

However, since nothing is known about the system in advance, gradient information is usually unavailable. This requires a method to obtain a gradient estimate $\boldsymbol{g}$. Since we do not pursue a uniform distribution of samples over the input space, or any other pattern, traditional indirect gradient estimation methods such as finite differences are inapplicable [13].

For each sample $\boldsymbol{p}_r \in P$ (referred to as the *reference sample*), a subset known as the neighbourhood is defined: $N(\boldsymbol{p}_r) = \{\boldsymbol{p}_{r1}, \ldots, \boldsymbol{p}_{rv}\} \subset P_r$ with $P_r = P \setminus \boldsymbol{p}_r$ and $v \geq 2d$. Least squares is applied on $N(\boldsymbol{p}_r)$ to obtain a gradient estimation $\boldsymbol{g}$. The difference between the true output value and the predicted value (using the gradient estimation) at neighbouring points is used as a measure of linearity. This measure is combined with an exploration based measure (provided by Voronoi) indicating how well the space surrounding the reference sample has been sampled. This information is used to rank all samples in P, and candidate samples are selected in the neighbourhood of the highest ranked samples. The key issue here is: how to define the neighbourhood to obtain a reliable gradient estimation based on the information available?

The LOLA algorithm answers this question with a *selective* approach, for a sample each possible neighbourhood of v points is scored in terms of two criteria:

1) *Cohesion*: A neighbour should be as close to the reference point as possible, as we are constructing a local approximation (proximity).
2) *Adhesion*: The neighbours should be as far away from each other as possible, in order to cover the space surrounding the reference point (surrounding).

Clearly it is impossible to maximize both: when the neighbourhood is very close to the reference point, it will have a higher adhesion value. The best trade-off for each sample (given the other samples) is saved. When N_{new} new samples have been evaluated and the algorithm is ran to determine new candidates for evaluation, each existing neighbourhood is revisited and updated if a better configuration is possible in addition to constructing N_{new} neighbourhoods. Although some optimizations exist to reduce the search space, the overall complexity of this approach is $\mathrm{O}\left(2^{2d} N N_{new}\right)$. When the dimensionality of the problem at hand increases, the run time increases dramatically.

III. Fuzzy Gradient Estimation

In this section a new approach is introduced to estimate the gradient in any $\boldsymbol{p}_r$. The method has a complexity of only $\mathrm{O}\left(N\right)$, leading to much faster runtime. Instead of carefully selecting v points as neighbourhood, all points within a radius α are included in the neighbourhood:

$$N(\boldsymbol{p}_r) = \{\boldsymbol{p} | \boldsymbol{p} \in P_r, ||\boldsymbol{p} - \boldsymbol{p}_r||_2 < \alpha\}.$$

To be able to solve a Least Squares problem the cardinality of $N(\boldsymbol{p}_r)$ has to be atleast d, if this requirement is not met, the system becomes underdetermined: this is the only constraint on the choice of α.

With this definition of the neighbourhood, the notion of *cohesion* and *adhesion* is lost. To reintroduce this valuable information, we formally define these two concepts $\forall \boldsymbol{p} \in N(\boldsymbol{p}_r)$:

$$C(\boldsymbol{p}_r, \boldsymbol{p}) = ||\boldsymbol{p} - \boldsymbol{p}_r||_2,$$

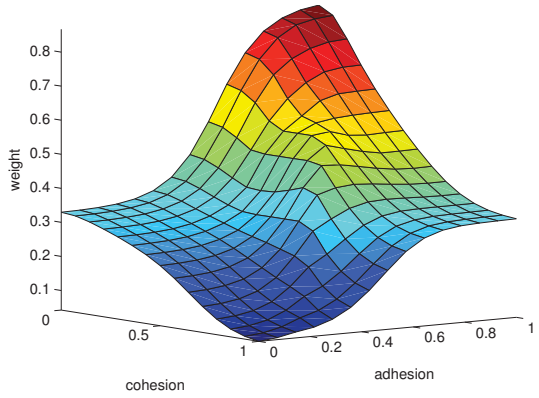
$$A(\boldsymbol{p}_r, \boldsymbol{p}) = \min_{\boldsymbol{q} \in P_r} ||\boldsymbol{q} - \boldsymbol{p}||_2.$$

RULE	COHESION	ADHESION	WEIGHT
1	CLOSE	LOW	HIGH
2	CLOSE	HIGH	AVERAGE
3	NOT CLOSE	LOW	AVERAGE
4	NOT CLOSE	HIGH	LOW

TABLE I. FIS RULE BASE FOR WEIGHT ASSIGNMENT.

These values are used to assign a weight for each point in the neighbourhood by using them as inputs to a Mamdani *Fuzzy Inference System* (FIS) [14]. For cohesion, a single membership function *close* is defined. The adhesion input parameter has two membership functions are defined: *low* and *high*. The weight output range is $[0, 1]$ and is subdivided in 3 membership functions: low, average and high. The relation between the input variables and the weight is described by four IF THEN rules as listen in Table I. The weights are then used to solve a Weighted Least Squares with the points in $N(\boldsymbol{p}_r)$ to obtain the gradient estimate $\boldsymbol{g}$. An example of the response of the FIS is shown in Fig. 3.

Fig. 3. Example of the response of the FIS. Points with high cohesion (= low numerical values, as it represents the distance to the reference point) and low adhesion strongly influence the gradient estimation, as opposed to points with low cohesion and high adhesion.



Because this gradient estimation method does not have to score several candidate neighbourhoods but estimates it directly from the surrounding space, a lot of computational complexity is avoided. The algorithm mainly relies on distance calculations, which can be done efficiently by calculating a distance matrix prior to estimating the gradient in each sample. The gradient estimation can occur in parallel, since the computation is independent for each sample. When new candidates have been evaluated by the simulator and added to the set P, the distance matrix can be updated in the next sequential design iteration. This means the complexity of LOLA with Fuzzy gradient estimation (FLOLA) only grows in terms of the size of the set P (which is N).

IV. NUMERICAL EXPERIMENTS

The FLOLA algorithm can be combined with a Voronoi exploration algorithm, similar to LOLA-Voronoi. This new algorithm will be referred to as *FLOLA-Voronoi*. The experiments are performed with the SUrrogate MOdeling (SUMO) MATLAB toolbox [15]. Designed as a research platform for sequential sampling and adaptive surrogate modeling featuring high extensibility, this MATLAB toolbox makes it very easy to implement and compare this new sampling approach to other sequential design methods. The work-flow of the SUMO Toolbox is illustrated in Fig. 1.

TABLE II. SUMMARY OF THE RESULTS FOR MODELING TEST CASE 4. THE LANGERMANN FUNCTION WAS SAMPLED UP TO 1500 POINTS, AND IT'S RRSE OVER A PRE-EVALUATED VALIDATION SET OF THE FINAL MODEL WAS RECORDED FOR SEVERAL MODEL TYPES. EACH EXPERIMENT RAN TEN TIMES TO CANCEL OUT NOISE BY RANDOM FACTORS. THE 95% CONFIDENCE INTERVALS ARE SHOWN BETWEEN BRACKETS.

	LS-SVM	RBF	ANN
FLOLA-Voronoi	0.32 (0.31, 0.33)	0.18 (0.17, 0.19)	**0.23** (0.21, 0.25)
LOLA-Voronoi	**0.31** (0.30, 0.32)	0.18 (0.17, 0.19)	**0.23** (0.22, 0.24)
Voronoi	0.35 (0.34, 0.36)	0.24 (0.23, 0.25)	0.27 (0.25, 0.29)
Model Error	0.61 (0.53, 0.68)	**0.17** (0.16, 0.18)	0.63 (0.55, 0.71)
Random	0.49 (0.46, 0.53)	0.37 (0.35, 0.39)	0.37 (0.35, 0.39)

A correct model evaluation process is crucial to the success of the surrogate modelling process [16]. For the experiments in this section, the Root Relative Square Error was calculated over a dense pre-evaluated validation set:

$$\mathrm{RRSE}(x, \tilde{x}) = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \tilde{x_i})^2}{\sum_{i=1}^{N} (x_i - \bar{x})^2}}.$$

x_i represents the vector of the true responses at all samples, $\tilde{x_i}$ represents the estimate by the surrogate model, and $\bar{x}$ the mean. Next to FLOLA-Voronoi and LOLA-Voronoi, three more sequential designs were tested: the exploration-based Voronoi component (without additional exploitation algorithm), a Model Error based sampling approach which samples in regions where the intermediate models make mistakes, and random sampling.

A. Langermann 2D

The Langermann function is very complex: it features several highly non-linear regions spread over the input space, with quasi-flat surfaces between them. Modelling this function up to a RRSE of 0.05 can be very time consuming: model types such as Kriging will suffer from numerical instability as many samples are required to obtain this level of accuracy. An initial Latin hypercube of 46 points complemented with a 2-level factorial design is used as a starting point. Each iteration, a batch of 25 points is added by the sequential design strategy and three types of models are constructed: Radial Basis Functions (RBF), Least-Squares SVM's [17] optimized with the DIRECT algorithm [18], and Artificial Neural Networks (ANN) trained with Levenberg-Marquardt backpropagation with Bayesian regularization (300 epochs) [19]. The topology and initial weights of the neural networks are optimized by a genetic algorithm. This process is repeated up to a maximum of 1500 samples.

Table II shows the results: with the exception of the combination of Model Error and RBF models, the best final scores are obtained with LOLA-Voronoi and FLOLA-Voronoi. The scores obtained with both methods are similar, which indicates that the new exploitation based algorithm (FLOLA) behaves like its predecessor.

This observation is confirmed by the evolution of the error as the modeling process evolves and more samples are

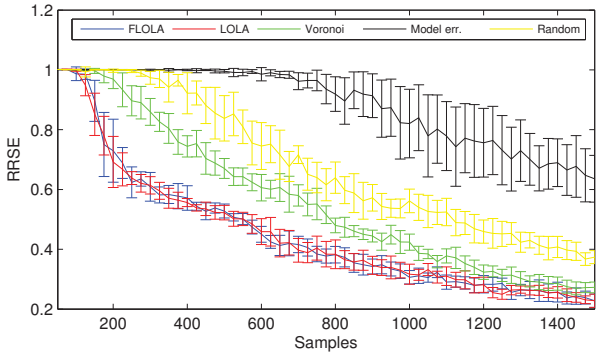


Fig. 4. Evolution of the average RRSE in function of the sample size for the Langermann function modeled with Artificial Neural Networks. The evolution of the error of FLOLA-Voronoi and LOLA-Voronoi is very similar and outperforms the other methods.

selected. Fig. 4 shows how the average error of the best model evolves for each sampling algorithm used to model the Langermann function with the ANN models. The 95% confidence intervals are indicated by the errorbars. It is clear that the FLOLA-Voronoi and LOLA-Voronoi algorithms perform very similar and outperform the other strategies.

B. Levy 4D

The second experiment a four dimensional Levy function: this is a medium-dimensional problem for global surrogate modelling. Because this function is less complex, it is possible to reach a strict target accuracy in reasonable time. Therefore, the experimental setup was changed: A Latin hybercube initial design of 200 points is created with the Translational Propagation (TPLHD) [20]. Sequentially, 50 points are added each iteration. As model type, Least-squares Support Vector Machines (LS-SVM) [17] optimized with the DIRECT algorithm [18] were used. Runs were terminated when a RRSE of 0.05 was reached, or a time limit of 24 hours was reached. Experiments were repeated ten times to cancel out the effect of random factors in the SUMO toolbox (for instance, in hyperparameter optimization). Next to the mean sample size, Table III also shows the 95% confidence intervals.

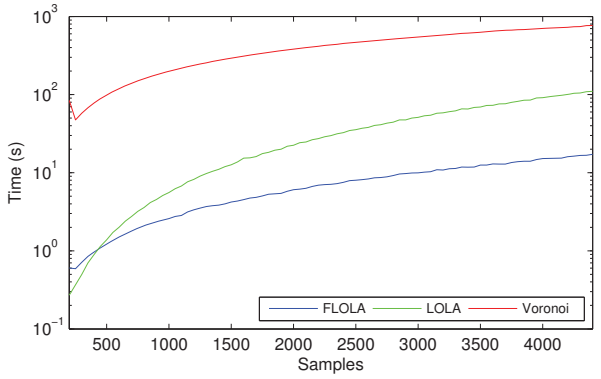


Fig. 5. Average runtime of the FLOLA, LOLA and Voronoi components.

Not a single run using the model error sequential design managed to reach the target accuracy within the time limit.

TABLE III. SAMPLE SIZES REQUIRED TO REACH A RRSE OF 0.05 FOR A 4D LEVY FUNCTION.

METHOD	RESULT
FLOLA-VORONOI	4715 (4625, 4804)
LOLA-VORONOI	4975 (4867, 5083)
VORONOI	5075 (4967, 5183)
MODEL ERROR	> 7878
RANDOM	7770 (7497, 8042)

On average, the runs were halted after 7878 samples, with an average accuracy of 0.08. This is very poor: even random sampling performs better. For random sampling, three runs did not complete because they exceeded the available time. These runs were not repeated but excluded, resulting in wider confidence intervals. The best results are obtained with FLOLA-Voronoi, which scores slightly better compared to its predecessor. Compared to only using Voronoi there is a difference in number of samples, but it is not very explicit as the Levy function does not feature strong non-linearity. However the FLOLA component of the algorithm is faster than Voronoi itself as shown in Fig. 5, and a lot faster than LOLA. Even in high-dimensional cases, including FLOLA as exploitation component can improve the modeling results at a small cost, which is not the case for the LOLA algorithm.

V. CONCLUSION

The proposed Fuzzy gradient estimation method greatly enhances the LOLA algorithm towards much faster sample selection with the FLOLA-Voronoi algorithm, without affecting the quality. Numerical experiments show the amount of samples required to reach the target accuracy remains equivalent or better, which is considerably less compared to other sequential design methods used in this study. For a medium-dimensional problem the difference in runtime between the old and new approach is significant, because the original gradient estimation method in LOLA has an exponential complexity in terms of the dimensionality. This motivates our choice to further investigate FLOLA-Voronoi for global surrogate modeling of high-dimensional problems.

In further work, different methods for automatically adapting the α parameter will be investigated. Furthermore the current algorithm chooses a new candidate in the neighbourhood of the samples highest ranked by the FLOLA-Voronoi algorithm based on the maximum minimum distance to other samples. Better options to improve this local space-fillingness will be investigated as well.

ACKNOWLEDGMENT

This research has (partially) been funded by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office. Experiments were carried out using the STEVIN Supercomputer Infrastructure at Ghent University, funded by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation and the Flemish Government department EWI. Dirk Deschrijver is a postdoctoral researcher of FWO-Vlaanderen.

REFERENCES

[1] D. Gorissen, W. Hendrickx, K. Crombecq, and T. Dhaene, "Adaptive Distributed Metamodeling," in *Proceedings of 7th International Meeting on High Performance Computing for Computational Science (VECPAR 2006), Springer - Lecture Notes in Computer Science*, M. Dayde *et al.*, Eds., vol. 4395, 2007, pp. 579–588.

[2] D. Deschrijver, B. Gustavsen, and T. Dhaene, "Advancements in iterative methods for rational approximation in the frequency domain," *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1633–1642, 2007.

[3] W. Van Beers, "Kriging metamodeling in discrete-event simulation: an overview," in *Proceedings of the 37th conference on Winter simulation*. Winter Simulation Conference, 2005, pp. 202–208.

[4] L. Balewski and M. Mrozowski, "Creating neural models using an adaptive algorithm for optimal size of neural network and training set," in *15th International Conference on Microwaves, Radar and Wireless Communications, MIKON-2004*, vol. 2, May 2004, pp. 543–546, conference proceedings of MIKON 2004.

[5] A. Lamecki, P. Kozakowski, and M. Mrozowski, "CAD-model construction based on adaptive radial basis functions interpolation technique," vol. 3, pp. 799–802, 2004, microwaves, Radar and Wireless Communications, 2004. MIKON-2004.

[6] K. Crombecq, D. Gorissen, L. D. Tommasi, and T. Dhaene, "A Novel Sequential Design Strategy for Global Surrogate Modeling," in *Proceedings of the 41th Conference on Winter Simulation, Austin, TX*, Dec. 2009, pp. 731–742.

[7] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A Novel Hybrid Sequential Design Strategy for Global Surrogate Modelling of Computer Experiments," *SIAM Journal of Scientific Computing*, vol. 33, no. 4, pp. 1948–1974, 2010.

[8] D. Deschrijver, F. Vanhee, D. Pissoort, and T. Dhaene, "Automated near-field scanning algorithm for the emc analysis of electronic devices," *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 3, pp. 502–510, 2012.

[9] P. Singh, D. Deschrijver, D. Pissoort, and T. Dhaene, "Accurate hotspot localization by sampling the near-field pattern of electronic devices," 2013.

[10] D. Deschrijver, K. Crombecq, H. M. Nguyen, and T. Dhaene, "Adaptive sampling algorithm for macromodeling of parameterized-parameter responses," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 1, pp. 39–45, 2011.

[11] D. Gorissen, D. Deschrijver, T. Dhaene, and D. D. Zutter, "A software framework for automated behavioral modeling of electronic devices [application notes]," *IEEE Microwave Magazine*, vol. 13, no. 6, pp. 102–118, 2012.

[12] S. Aerts, D. Deschrijver, W. Joseph, L. Verloock, F. Goeminne, L. Martens, and T. Dhaene, "Exposure assessment of mobile phone base station radiation in an outdoor environment using sequential surrogate modeling," *Bioelectromagnetics*, vol. 34, no. 4, pp. 300–311, 2013.

[13] M. C. Fu, "Gradient estimation," *Handbooks in operations research and management science*, vol. 13, pp. 575–616, 2006.

[14] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[15] D. Gorissen, K. Crombecq, I. Couckuyt, P. Demeester, and T. Dhaene, "A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, 2010, available at http://sumo.intec.ugent.be. [Online]. Available: http://sumo.intec.ugent.be

[16] D. Gorissen, I. Couckuyt, E. Laermans, and T. Dhaene, "Multiobjective global surrogate modeling,dealing with the 5-percent problem," *Engineering with Computers*, vol. 26, no. 1, pp. 81–89, Jan. 2010.

[17] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel, *Least squares support vector machines*. World Scientific, 2002, vol. 4.

[18] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[19] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[20] F. A. Viana, G. Venter, and V. Balabanov, "An algorithm for fast optimal Latin hypercube design of experiments," *International journal for numerical methods in engineering*, vol. 82, no. 2, pp. 135–156, 2010.