ON COMBINING SEVERAL OPTIMIZATION ALGORITHMS FOR IMPROVING THE EFFICIENCY OF IC VERIFICATION PROCESS

Alecsandra RUSU, Ingrid KOVACS, Bianca CĂRBUNESCU, Marina ȚOPA, Andi BUZO, Georg PELZ, Emilian DAVID

Technical University of Cluj-Napoca, Romania, Alecsandra.Rusu@bel.utcluj.ro
Technical University of Cluj-Napoca, Romania, Ingrid.KOVACS@staff.utcluj.ro
Technical University of Cluj-Napoca, Romania, Bianca.Carbunescu@bel.utcluj.ro
Technical University of Cluj-Napoca, Marina.Topa@bel.utcluj.ro
Infineon Technologies AG, Neubiberg, Germany, Andi.Buzo@infineon.com
Infineon Technologies AG, Neubiberg, Germany, Georg.Pelz@infineon.com
Infineon Technologies, Bucharest, Romania, EmilianConstantin.David@infineon.com

Abstract: In the modeling of complex integrated circuits, the locations of the sampled data points are essential to the success of the verification process. Most of the adaptive sampling methods based on iterative algorithms, that use data acquired from previous iterations to guide future sample selection, are optimization algorithms. An essential consideration of every adaptive sampling method is the trade-off between exploration and exploitation, i.e., the capability of the method to cover the under-sampled parts of the design space, but also focusing on interesting regions. This paper proposes three combinations of existing methods with the purpose of minimizing this trade-off and efficiently characterizing the verification space. Their performance and efficiency were validated on 20 synthetic test functions.

Keywords: *IC verification, adaptive sampling, optimizer.*

I. INTRODUCTION

Nowadays, one of the most technologically advanced industries is the semiconductor manufacturing industry. Integrated circuits began to be produced in very fine technologies and they started to include increasingly complex functions [1]. In view of the complexity in semiconductor industry, as well as the increasing demand for faster designs with growing quality requirements, finding methods that greatly increase the speed of evaluating system performance becomes crucial in order to maintain the company’s profitability [2]. To comply with these requirements, accurate simulation models must be used, which must evaluate the performance of the system in the best determined simulation points of the entire permitted design space, which is not a trivial task, as the simulation of complex systems with multiple input and output parameters can be a time-consuming process. Furthermore, the simulation effort explodes, however, when the performance has to be analyzed for a continuous range of several parameter settings. Since simulations can be very expensive, the data points, which are required to build the model, must be chosen as optimally as possible [3].

Sequential optimization methods (used in adaptive sampling) analyze data (models and samples) from previous iterations conducive to select new samples in areas that are more difficult to approximate, resulting in a more efficient distribution of samples compared to traditional design of experiments [4]. In a representative adaptive sampling method, firstly, an initial set of samples is evaluated using a minimal experimental design. Then a model is built using this data and based on the estimated accuracy of the model, the algorithm may decide if more samples are required [5]. Figure 1 depicts the adaptive sampling flow.

Verification in IC process is essentially an optimization problem where the aim is to find the worst-case behavior in relation to the performance within the limits of specification. This also comes with ensuring certainty, in terms of the minimum performance found by exploring as completely as possible the space of depending factors. By factors we refer to any of the following: system parameters, operating conditions, production process parameters, while an example of system response is the power consumption. While the number of evaluations must be kept to a minimum value to avoid rising costs.

The traditional optimization algorithms come with certain difficulties in achieving this ideal compromise in terms of exploration-exploitation, so combining them can bring an extra benefit in the efficiency of verification. In this paper, we will prove that it is so.

An essential consideration in adaptive sampling is the trade-off between global exploration and local exploitation. Exploitation-based methods estimate the error of the model over the design space and select new samples in locations where the estimated error is the largest. Exploration-based methods, on the other hand, try to improve the domain coverage by selecting samples in such a way that the design space is covered as uniformly as possible [6]. The best optimization method is the one that

Manuscript received December 2, 2021; revised January 13, 2022

manages to combine intelligently global exploration and local exploitation.

In this paper, we propose three new hybrid sequential optimization methods to improve the exploration-exploitation trade-off. In this way, we overcome the disadvantage of the Uncertainty reduction adaptive sampling method [7], mainly used for exploration purposes, by combining it with suitable optimizers for local exploitation purposes. Thus, the global exploration role will be fulfilled by Uncertainty reduction and in terms of local exploitation; optimization algorithms such as Differential Evolution [8], Fmincon [9], and Pattern search [10] have proven to be a very good choice.

The paper is organized as follows: in Section II we present the problem formulation and provide a brief description of the methods used in this paper; in Section III, we describe the setup of the new methods, generated as a combination between the Uncertainty reduction and an optimizer: Uncertainty reduction & Differential Evolution, Uncertainty reduction & Fmincon and Uncertainty reduction & Pattern search.; in Section IV we demonstrate the effectiveness of the new methods to cover the entire analysis space of the test functions and, at the same time, accurately determine the regions of interest (extremes in this case); conclusions are drawn in Section V.

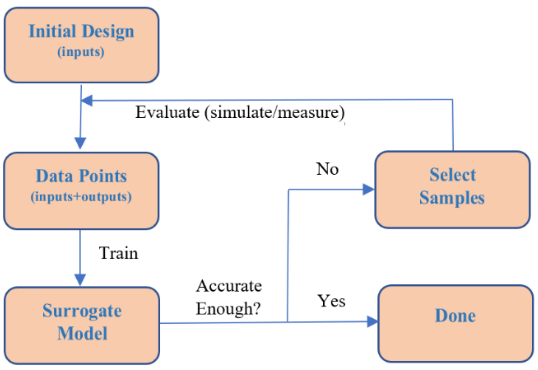


Figure 1. An adaptive sampling flowchart.

II. PROBLEM FORMULATION

2.1 Context

An essential consideration in adaptive sampling is the trade-off between exploration and exploitation. If an adaptive sampling strategy focuses only on local exploitation, the initial experimental design must be sufficiently large as to capture all regions of interest right away and avoid large (interesting) areas to remain unsampled. On the other hand, if a strategy focuses only on global exploration, the advantage provided by evaluating and selecting the samples sequentially is ignored, because the outputs are not used.

In order to increase the sampling performance, these two parts should be mixed in a balanced manner, which is conceptually expressed as:

$$Score = w_{local} \times \text{local} + w_{global} \times \text{global} \qquad (1)$$

where w_{local} and w_{global}, which are assumed to satisfy $w_{local} + w_{global} = 1$, are the weights for local exploitation and global exploration. Local and global represent the local exploitation term and global exploration term, respectively [11].

However, current adaptive sampling approaches use a fixed rule to balance local exploitation and global exploration. The SEED approach [12] adopts a balance factor λ; [13] presented a decreasing law of the weights according to several error thresholds; the LOLA-Voronoi approach [14] assigns the same weight for local exploitation and global exploration. [15] employed balance strategy to perform adaptive sampling by circularly looping through a search pattern that contains several weights from global to local exploitation.

We propose a novel, generic approach, in which two different criteria are defined: one for global exploration and one for local exploitation, we considered $w_{local} = w_{global}$=0.5. For the global exploration criterion, we have chosen an adaptive sampling method based on sampling in areas where metamodel prediction currently is most uncertain, Uncertainty reduction. For the local exploitation criterion, we used three optimization methods (Differential Evolution, Fmincon and Pattern search), see Figure 2.

2.2 Adaptive sampling and optimization approaches

Uncertainty Reduction Method

Uncertainty reduction method is an adaptive sampling technique, which samples the space in the region where the prediction currently is most uncertain (Figure 3). The algorithm starts with a small number of uniformly distributed samples as an initial training set to create a metamodel. Using the metamodel simulation, it gets the response of the new points from this uncertainty region. Then, it moves on to the next stage, training a new metamodel. This procedure is repeated until there are enough samples from simulations to get a sufficiently good fit of the response curve [7].

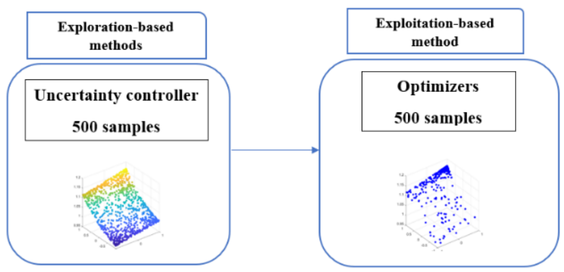


Figure 2. Schematic representation of the new adaptive sampling strategy.

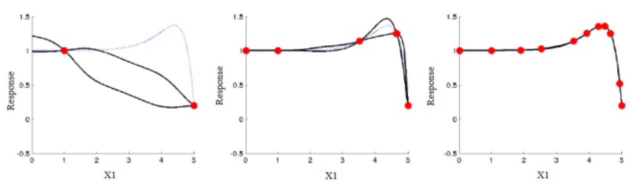


Figure 3. Concept of Uncertainty reduction method (the red points are the samples generated with Uncertainty reduction and the curve illustrates the shape of the simulated function) [7].

Differential Evolution Method
The differential evolution (DE) algorithm belongs to a broader family of evolutionary computing algorithms. It starts by randomly initiating a population of real-valued decision vectors, also known as genomes or chromosomes [8]. The mutations are performed on the population, at each iteration, to generate new candidate solutions (Figure 4). The mutation process adds the weighted difference between two population vectors to a third vector, to produce a mutated vector. The factors of the mutated vector are again mixed with the factors of another predetermined vector, the target vector, during a process known as crossover that aims to increase the diversity of the perturbed parameter vectors. The resulting vector is known as the trial vector. These iterations continue until a termination criterion is reached [16].

Fmincon Method
The software module fmincon is a solver provided by MATLAB used on both linear and nonlinear systems. The function fmincon is an optimizer that supports solving large, structured problems. Fmincon has 4 algorithm options that can be selected from the field options [9]. This function is a gradient based function and can be used to search and find all possible local extremes that satisfy the given objectives. The iteration process starts with an initial guess and stops when all setup criteria are met. If the first-order optimization is fulfilled by the last iteration, the result is considered as a local extreme that satisfies system needs (Figure 5) [17].

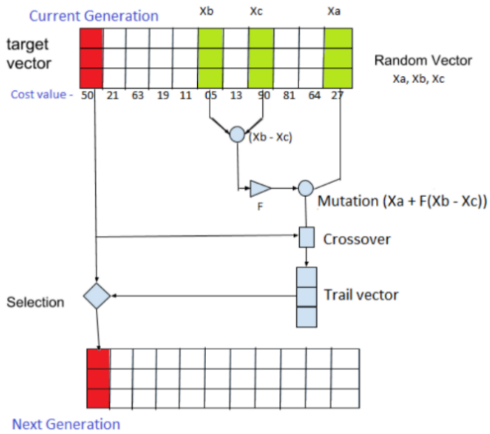


Figure 4. Concept of Differential evolution method [16].

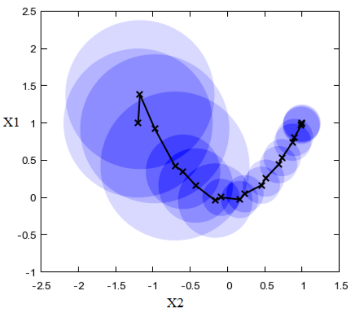


Figure 5. Trace of unconstrained optimization with fmincon [18].

Pattern search Method
Pattern search (also known as direct search, derivative-free search or black-box search) is a family of numerical optimization methods provided by MATLAB that does not require any information about the gradient of the objective function (Figure 6). This algorithm searches a set of points around the current point, looking for a lower value of the objective function than the value of the current point. Direct search can be used to solve problems for which the objective function is not differentiable or is not even continuous. Global Optimization Toolbox functions include three Pattern search algorithms called: the generalized pattern search (GPS) algorithm, the generating set search (GSS) algorithm and the mesh adaptive search (MADS) algorithm [10].

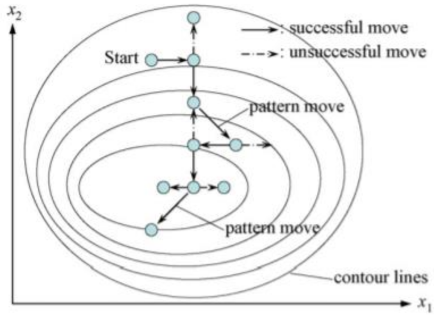


Figure 6. The concept of the pattern search method [10].

III. EXPERIMENTAL SETUP AND RESULTS

To test the existing and proposed methods let us consider a set of 20 synthetic test functions (monotonic functions and also functions with extreme points); these functions can be grouped into five categories as shown in Figure 7. The experiments were performed on these test functions with different number of factors; it started with 2 factors, it continued with 5 factors and finally 10 factors were considered for each function. In all cases, the factors were varied in the range [-1, 1].

The first experiments were performed on the Uncertainty reduction and independently on each of the optimizers.

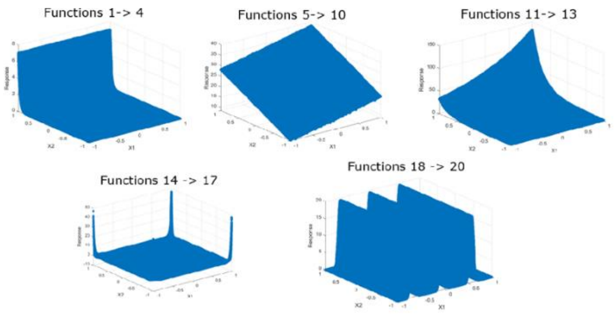


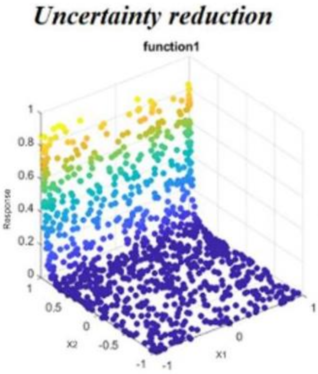
Figure 7. Test functions overview.

Within these experiments, a maximum number of 1000 samples were allocated for each of the tested methods. It was noticed that the first method offers a very good coverage of the entire space of the function but does not insist on the extreme areas. Meanwhile the optimizers insist on the extreme area, but do not offer a good coverage of the whole space.

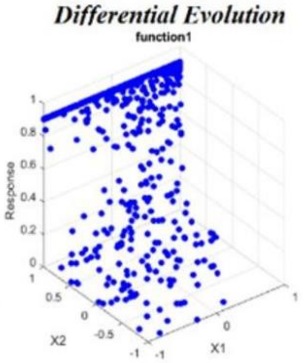
The second stage of the experiments consists of generating new methods by combining the Uncertainty reduction with each of the optimizers in two steps, as shown in Figure 2.

Uncertainty reduction & Differential Evolution method was obtained by considering the 500 samples generated by Uncertainty reduction. From these samples, the first 10 highest values were selected as the initial population in Differential Evolution. Taking into account the initial population set by us, Differential Evolution generated 500 samples. From Figure 8.a. and Figure 8.b. it can be understood that the methods worked as we expected, they cover the whole space and at the same time emphasizes the analysis on the extreme areas. But there is a disadvantage: the long experimental time due to the large number of required evaluations of functions.

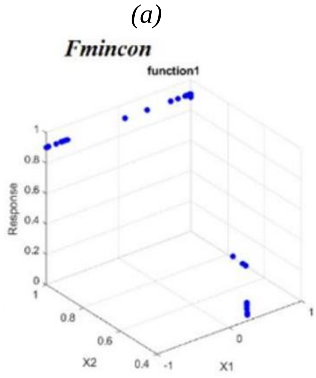
For the methods Uncertainty reduction & Fmincon and Uncertainty reduction & Pattern search, the 500 samples generated by Uncertainty reduction were also considered. From this set of 500 samples, the maximum value (which is identified as local exploitation term) was determined and was added as a starting point parameter for the two optimizers, Fmincon and Pattern search. For these optimization methods the maximum number of samples was set at 500, but they used less, see Table 1, Figure 8. c and Figure 8. d



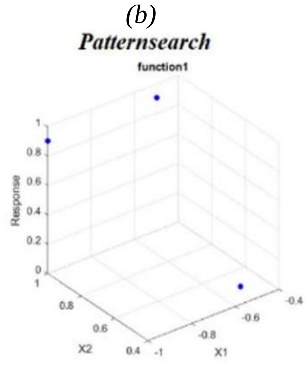
(a)



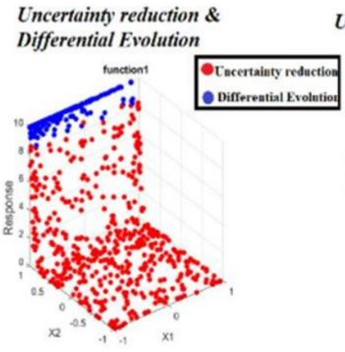
(b)



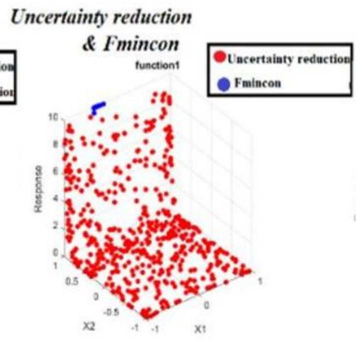
(c)



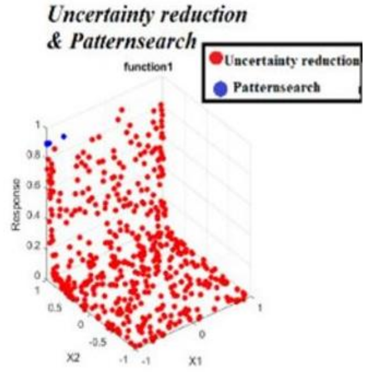
(d)



(e)



(f)



(g)

Figure 8. Data points for the following methods: (a) Uncertainty; (b) Differential Evolution; (c) Fmincon; (d) Pattern search; (e) Uncertainty reduction & Differential Evolution; (f) Uncertainty reduction & Fmincon; (g) Uncertainty reduction & Pattern search

The disadvantage of the Uncertainty reduction & Fmincon method, can be considered the risk of getting stuck around the starting point. The Uncertainty reduction & Pattern search method thus prove to be the most advantageous choice and with the fewest compromises.

3.2 Results

By visually comparing the figures 8.a and 8.b, it can be deduced that Uncertainty reduction offers the best coverage of the analysis space and the three optimizers insist in the extreme area.

To analyze how the local exploitation part is performed, the comparison of the obtained maximum values was used. We generated, for each function, 1000000 samples with Differential Evolution, determine the maximum value, then this maximum value is compared with the maximum value obtained with each of the three methods (Uncertainty reduction & Differential Evolution, Uncertainty reduction & Fmincon and Uncertainty reduction & Pattern search).

When the 20 test functions have 2 factors, all 3 methods behave very well, as confirmed in Table 1. However, the Uncertainty reduction & Fmincon and Uncertainty reduction & Pattern search methods have the lowest values for the cost of evaluating the functions. The methods worked as expected, they cover the whole space and at the same time forces the analysis on the extreme areas.

From the careful analysis of Table 2 and Table 3 it can be seen that Uncertainty reduction & Pattern search is the method with the best results, but also with the highest number of function evaluations. To have a clear comparison between the three methods, we tested the Uncertainty reduction & Differential Evolution method with the same number of function evaluation (500&600 in case of 5 factors and 500&1700 in case of 10 factors). From this test, two observations can be made:

- the best results are those obtained with the Uncertainty reduction & Pattern search method,
- With the increase of the number of factors the methods Uncertainty reduction & Differential Evolution need more samples to obtain conclusive results.

For an even more complex analysis on the methods, Tables 4-6 were filled in with the results obtained after comparing the maxima obtained by the methods with 1% error compared to the real maxima.

The results presented in the last 3 tables reinforce the previous observations: the method with the best results is the Uncertainty reduction & Pattern search method; the other two methods need a larger number of samples and a higher number of function evaluations to offer satisfactory results.

Table 1. Overview of the results obtained on the 20 functions of 2 factors

Controller type	Percentage of determined maxima	Function Evaluation	No. of samples
Uncertainty	35%	1000	1000
Differential Evolution	85%	1000	1000
Fmincon	60%	38-239	12-36
Pattern search	90%	47-218	22-72
Uncertainty & Differential	95%	500 & 500	500& 500
Uncertainty & Fmincon	95%	500 & (21-94)	500 & (5-25)
Uncertainty & Pattern search	95%	500 & (59-181)	500 & (30-60)

Table 2. Overview of the results obtained on the 20 functions of 5 factors

Controller type	Percentage of determined maxima	Function Evaluation	No. of samples
Uncertainty	0%	1000	1000
Differential Evolution	0%	1000	1000
Fmincon	55%	116-315	15-52
Pattern search	75%	359-680	90-138
Uncertainty & Differential	20%	500 & 500	500& 500
Uncertainty & Fmincon	45%	500 & (92-276)	500 & (16-37)
Uncertainty & Pattern search	75%	500 & (305-602)	500 & (92-124)
Uncertainty & Differential	65%	500 & 600	500 &600

Table 3. Overview of the results obtained on the 20 functions of 10 factors

Controller type	Percentage of determined maxima	Function Evaluation	No. of samples
Uncertainty	0%	1000	1000
Differential Evolution	0%	1000	1000
Fmincon	45%	187-918	17-46
Pattern search	90%	1242-1827	32-200
Uncertainty & Differential	0%	500 & 500	500& 500
Uncertainty & Fmincon	55%	500 & (156-276)	500 & (14-45)
Uncertainty & Pattern search	95%	500 & (1246-1693)	500 & (152-198)
Uncertainty & Differential	5%	500 & 1700	500 &1700

Table 4. Overview of the results obtained on the 20 functions of 2 factors (1% error)

Controller type	Percentage of determined maxima
Uncertainty	65%
Differential Evolution	85%
Fmincon	65%
Pattern search	90%
Uncertainty & Differential	100%
Uncertainty & Fmincon	100%
Uncertainty & Pattern search	100%

Table 5. Overview of the results obtained on the 20 functions of 5 factors (1% error)

Controller type	Percentage of determined maxima
Uncertainty	0%
Differential Evolution	40%

Fmincon	60%
Pattern search	90%
Uncertainty & Differential	55%
Uncertainty & Fmincon	65%
Uncertainty & Pattern search	90%

Table 6. Overview of the results obtained on the 20 functions of 10 factors (1% error)

Controller type	Percentage of determined maxima
Uncertainty	0%
Differential Evolution	0%
Fmincon	70%
Pattern search	95%
Uncertainty & Differential	10%
Uncertainty & Fmincon	75%
Uncertainty & Pattern search	100%

IV. CONCLUSION

The paper proposes novel, highly efficient and very robust generic adaptive sampling methods that minimize the exploration-exploitation trade-off of some well-known adaptive sampling methods.

Discovering that Uncertainty reduction manages to cover well the entire space of analysis, but does not insist enough on the extreme areas, we completed the shortcoming by combining it with an optimizer for exploitation purpose. The optimizers (Differential Evolution, Fmincon and Pattern search) will only insist on the extreme area. Thus, our combined methods (Uncertainty reduction & Differential Evolution, Uncertainty reduction & Fmincon and Uncertainty reduction & Pattern search) provide a complete analysis of the space, covering the entire analysis domain, including the extreme areas.

Balancing the cost and efficiency of the three methods, it was proved that Uncertainty reduction & Pattern search is the best method. This method requires a small number of samples and an acceptable cost in terms of time and function evaluations. This is especially noticeable in the case of 10-factor functions, where the method identified a maximum of 19 of 20 functions with only 698 samples and a number of function evaluations of 2193. However, the other two methods also manage to obtain a special balance between exploitation and global exploration, offering both the coverage of the entire space and the focus on the extreme area. The disadvantage of these two methods is the large number of required samples and a high number of function evaluations.

We can conclude that, the proposed method: Uncertainty reduction & Pattern search provides a good alternative for classical sequential experimental design techniques by reducing the total number of simulations requires. Future work includes testing and validating the proposed methods on real data sets from IC verification.

V. ACKNOWLEDGMENT

This work was co-funded by the European Regional Development Fund through the Operational Program "Competitiveness" POC -A1.2.3-G-2015, project P_40_437, contract 19/01.09.2016.

REFERENCES

[1] A. Rusu, I. Kovacs, B. Cărbunescu, M. Topa, A. Buzo, G. Pelz, "Electrical Parameters Prediction for Fab-to-Fab IC Product Migration", SIITME, accepted, 2021.
[2] I. M. Kovacs, "Multivariate Performance Analysis of Smart Power Integrated Circuits," PhD thesis, Technical University of Cluj-Napoca, 2018
[3] P. Westermann, R. Evins, "Adaptive Sampling for Building Simulation Surrogate Model Derivation Using The LOLA-Voronoi Algorithm," *Proceedings of the international building performance simulation association*, 16 (2019), pp. 1559-1563.
[4] D. C. Montgomery, "Design and Analysis of Experiments," New York: John Wiley, pp. 490-542, 2001
[5] T. McConaghy, K. Breen, J. Dyck, A. Gupta," Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide", Springer New York Heidelberg Dordrecht London,pp. 169-180, 2013.
[6] K. Crombecq, "Surrogate Modelling of Computer Experiments with Sequential Experimental Design", PhD thesis, University of Ghent 2011.
[7] M. Dobler, M. Rafaila, „Rapid Design Space Exploration of a State-of-the-art PSI 5 Controller," MBMV, Neubiberg, 2013.
[8] A. H. a Rashid, Y. W. Choon, M. S. Mohamad, "Producing Succinic Acid in Yeast using A Hybrid of Differential Evolution and Flux Balance Analysis", International Journal of Bio-Science and Bio-Technology, pp. 5(6):91-100, December 2013.
[9] S. Boyd, L. Vandenberghe, "Convex Optimization", Cambridge University Press, Cambridge, New York, 2004.
[10] K. Fei, L. Junjie, "Artificial bee colony algorithm and pattern search hybridized for global optimization", Applied Soft Computing, pp. 1781-1791, April 2013.
[11] Liu, H., Ong, YS. & Cai, J., "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design", Struct Multidisc Optim, pp. 393–416, 2018.
[12] Y. Lin, "An efficient robust concept exploration method and sequential exploratory experimental design.", Georgia Institute of Technology, Atlanta, USA, 2004.
[13] B. Kim, Y. Lee, D-H Choi, "Construction of the radial basis function based on a sequential sampling approach using cross-validation.', Journal of mechanical science and Technology, 23(12), 3357-3365.
[14] K. Crombecq, D.Gorissen, D. Deschrijver, T.Dhaene, "A novel hybrid sequential design strategy for global surrogate modeling of computer experiments.", SIAM Journal on Scientific Computing 33 (4), 2011.
[15] H. Liu, S. Xu, Y. Ma, X. Chen, X. Wang, "An adaptive Bayesian sequential sampling approach for global metamodeling.", ASME. J. Mech. Des. January 2016.
[16] N. Ferrante, T. Ville, "Recent advances in differential evolution:", Springer Science+Business Media B.V, 2009.
[17] L.Hei, "Practical techniques for nonlinear optimization", Ph D, Northwestern University, Evanston, 2007.
[18] (2014), Trust-region methods, https://optimization.mccormick.northwestern.edu/index.php/Trust-region_methods.