Geostatistical-Inspired Fast Layout Optimization of a Nano-CMOS Thermal Sensor

Oghenekarho Okobiah[1], Saraju P. Mohanty[2], and Elias Kougianos[3]
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu)
Computer Science and Engineering[1,2] and Engineering Technology[3]
University of North Texas, Denton, TX 76207, USA.
Email ID: oo0032@unt.edu[1], saraju.mohanty@unt.edu[2], and elias.kougianos@unt.edu[3]

Abstract

Continuous and aggressive scaling of semiconductor technology has led to persistent and dominant nanoscale effects on analog/mixed-signal (AMS) circuits. Design space exploration and optimization costs using conventional techniques have increased to infeasible levels. Hence, growing research for alternative design and metamodeling techniques with a much reduced design space exploration and optimization cost and high level of accuracy, continues to be very active. Metamodeling techniques aim to replicate simulation results of analog solvers without incurring expensive simulation costs. This paper presents a geostatistical inspired metamodeling and optimization technique for fast and accurate design optimization of nano-CMOS circuits. The design methodology proposed integrates a simple Kriging technique with efficient and accurate prediction characteristics as the metamodel generation technique. A Gravitational Search Algorithm (GSA) is applied on the generated metamodel (substituted for the circuit netlist) to solve the design optimization problem. The proposed methodology is applicable to AMS circuits and systems. Its effectiveness is illustrated with the optimization of a 45 nm CMOS thermal sensor. With 6 design parameters, the design optimization time for the thermal sensor is decreased by 90 % and produces an improvement of 36.8 % in power consumption. To the best of the authors' knowledge *this is the first work to use GSA for analog design optimization*.

Keywords

Nano-CMOS, Geostatistics, Kriging Methods, Gravitational Search Algorithm, AMS Design Flow, Physical Design Optimization.

I. Introduction

The drive towards aggressive scaling of semiconductor technologies is fueled by market trends for smaller, more powerful and yet efficient electronic devices. However, analog/mixed-signal (AMS) circuit designs in the deep nanometer region are susceptible to, and even dominated by various nanoscale effects and process variation. With all of these effects, the number of design points and process parameters required for accurate simulation has become excessive and this makes exhaustive simulation of design models computationally intensive. Thus the research for alternative methods to alleviate this problem continues to be very active. Current research techniques used to reduce simulation time include the use of metamodeling functions [4, 3, 2, 1] and performance estimation through Monte Carlo (MC) simulations.

Metamodeling techniques aim to replicate the simulation results of CAD tools without incurring expensive simulation costs. Metamodels are usually generated by functions which are generally approximations of the performance objectives [1]. Some of the most common metamodeling techniques include low-order polynomial functions and artificial neural network models. While the goal of using metamodels is to reduce the expensive computational costs, the generated metamodels also have to be accurate enough to ensure efficient optimizations. The accuracy and efficiency of the generated metamodels is thus an important factor in their use for simulation and depends on the technique used in creating it [4]. For instance, metamodeling techniques based on low-order polynomial regression functions produce accurate descriptions, but perform inefficiently when used for global design optimizations [2]. In predicting the objective function, regression models assume the effects of process variation are purely random and approximate the error equally across the design space. However, in nano-CMOS and other technologies, this is not the case. These effects are not purely random, but are strongly correlated. Kriging based metamodels which are based on geostatistical techniques, take into account by their weighting system the correlation between the process parameters. Prediction based on Kriging can thus provide a robust metamodel which is process variation and yield aware. For designs with many parameters, a characteristic of many nano-CMOS circuits, the design space is very large and increases exponentially with problem size, making exhaustive search techniques impossible [5]. Optimization algorithms are used with the design models to solve the design optimization problem. Common optimization algorithms utilized for circuit design include genetic algorithms, swarm algorithms, simulated annealing, tabu search and geometric programming [6, 7, 4].

This paper presents geostatistical inspired metamodeling and optimization techniques for the fast and accurate design optimization of nano-CMOS circuits. The methodology proposed integrates a simple

Kriging based technique as the metamodel generation method. A Gravitational Search Algorithm (GSA) is applied on the metamodel to solve the design optimization problem. The proposed methods are applicable to AMS circuits and their effectiveness is illustrated here with the optimization of a 45 nm CMOS thermal sensor with power minimization as an objective.

The rest of this paper is organized as follows. The major contributions of this paper are outlined in Section II. A summary of related research is presented in Section III. The proposed design flow in presented in Section IV. In Section V, a brief overview of simple Kriging modeling is discussed. In Section VI, the background and theory of the Gravitational Search Algorithm is described. In Section VII, an illustration of the design flow is presented on the design optimization of a 45 nm thermal sensor. Conclusions and future research directions are presented in Section VIII.

II. Novel Contributions of This Paper

Minimization of power consumption in thermal sensors is very important. Such sensors are used to monitor the thermal level of circuits to increase reliability; undue high power consumption by the sensor only burdens the circuits they monitor, thus leading to increased temperatures that further decrease thermal reliability. The challenge for designers in the minimization of the power consumption is a tradeoff with the accuracy and sensitivity of the circuit. The proposed design methodology is used for the optimization of power consumption with thermal sensitivity as constraint.

In this paper, the use of Kriging methods is introduced in a design flow methodology for AMS design optimization. Kriging techniques provide accurate response predictions and are effective for processes with correlation effects; thus they can account for correlation effects. Simple Kriging is used for the ultra fast generation of layout-aware accurate metamodels. The design metamodel is then optimized with the gravitational search algorithm (GSA) that employs both exploitative and explorative aspects of population based algorithms using gravity rules. The GSA algorithm, developed in [5], is presented here for optimization of AMS circuits. The efficiency of the overall proposed methodology is illustrated using a 45 nm thermal sensor case study circuit.

The **novel contributions of this paper** to the state-of-the-art are the following:

1) A novel ultra-fast but accurate layout design optimization flow for AMS components that incorporates layout-aware metamodels and fast algorithms.
2) A layout-accurate method for Kriging metamodel generation of AMS blocks. As a specific example, the “simple Kriging method” is presented.
3) A novel Gravitational Search Algorithm (GSA) based layout optimization for AMS blocks.

4) As a specific case study, a 45 nm CMOS based thermal sensor power minimization is performed with thermal sensitivity as constraint.

III. Related Prior Research

Research for design exploration and simulation using metamodeling has been growing and still remains an important research topic. With the increasing complexity of nano-CMOS designs and associated costs for design and manufacturing technologies, the use of efficient and accurate metamodels is essential for reducing the design computational cost. Common metamodeling techniques include polynomial regression, artificial neural networks (ANN) and radial basis functions. A review of these methods is presented in [8, 9]. The most common metamodeling technique is based on low-order polynomials which evolved from the Design of Experiments (DOE) methodology [9]. Such metamodels provide fast and efficient locally accurate results but do not perform well in global optimization problems [2, 10]. Another metamodeling technique is presented in [7]; the metamodels are generated from the application of geometric programming to polynomial equations deduced from circuit designs. However, this method's accuracy is low due to the approximating equations and ignores parasitics.

Metamodels based on ANNs have been used in [10] for the modeling of discrete stochastic systems. Techniques to improve the selection of ANN structures are presented in [10]. The use of Kriging for circuit design has been researched in [11, 12]. In [1, 2], studies on Kriging metamodeling for stochastic simulations has been presented. Recently in [14, 13] a study of different Kriging methods, simple and ordinary, is presented. Fig. 1 shows a taxonomy of different metamodeling techniques.

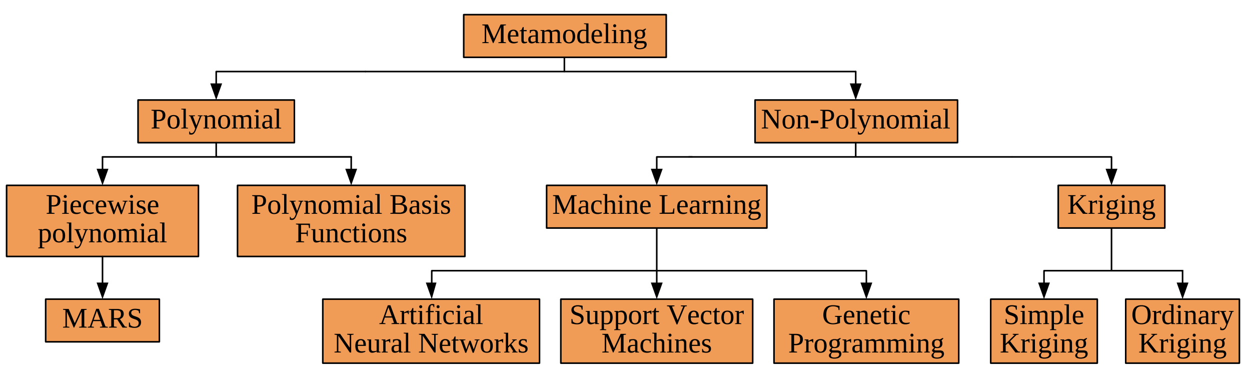


Fig. 1: A classification of metamodeling techniques.

A comparison of well known optimization algorithms including simulated annealing, genetic algorithms (GA) and gradient algorithms is presented in [6]. In [15], orthogonal optimization techniques based on swarm intelligence are presented. Recent research trends steer towards improving optimization time and efficiency, and multi-objective optimization [16, 17, 18]. Conventional algorithms include GA, swarm intelligence, simulated annealing, tabu search, gradient algorithms, linear and geometric programming [6, 7, 15, 17]. Evolutionary algorithms which operate heuristically are particularly suited for computational functions and achieve near-optimal solutions. The gravitational search algorithm (GSA) was recently proposed in [5] and its performance is comparable to particle swarm optimization.

The optimization of thermal sensor design is also a well researched topic. The need for low power designs without degrading the accuracy of temperature estimation poses a problem. A popular topology for thermal sensors is that of the ring oscillator [19, 20]. A design based on differential ring oscillators (DRO) was proposed in [19]. In [21], a methodology which incorporates statistical techniques into the design process aids the estimation of temperature effects on the circuit. In [22], a design which uses a reference transistor independent of ambient temperature is proposed. The effects of noise and process variation are modeled into the temperature reading increasing accuracy.

IV. The Proposed Design Optimization Flow

An overview of the proposed design optimization methodology is given in Fig. 2. The design process integrates a geostatistics-inspired (simple Kriging) parasitic-aware metamodel which is optimized with the gravitational search algorithm. The first step is the circuit schematic to meet design specifications. A common specification for most thermal sensors is a sensitivity of $0.05 \sim 1$ °C [23]. Once the schematic design is complete, the physical layout design is performed. A full blown parasitic netlist (R-resistance, L-self inductance, C-capacitance, and K-mutual inductance) is extracted from the layout and used for further simulations to ensure silicon-aware accuracy. The parasitic netlist is parameterized with the design variables. The parameterization of the netlist enables the easy generation of sample point locations from the physical layout used in the metamodel generation. The thermal sensor design netlist is parameterized for 6 variables. The next major step is the creation of the metamodel. Sample points are generated using Latin Hypercube Sampling (LHS) and are used as an input to the metamodel generator. Details of the metamodel generation are presented in Section V. To optimize the design, the optimization algorithm is used on the created metamodel for intelligently solving the design objective problem. The input to the optimization algorithm is the metamodel and the output is the optimized design parameter points that yield an optimal solution. A background for the GSA algorithm used and detailed flow of the optimization algorithm are given in Section VI. The final step is to resize the physical design using the optimal design parameters. The use of parameterized netlists limits the number of manual physical design iterations to one and ensures an optimal design which is parasitic aware as well as optimal.

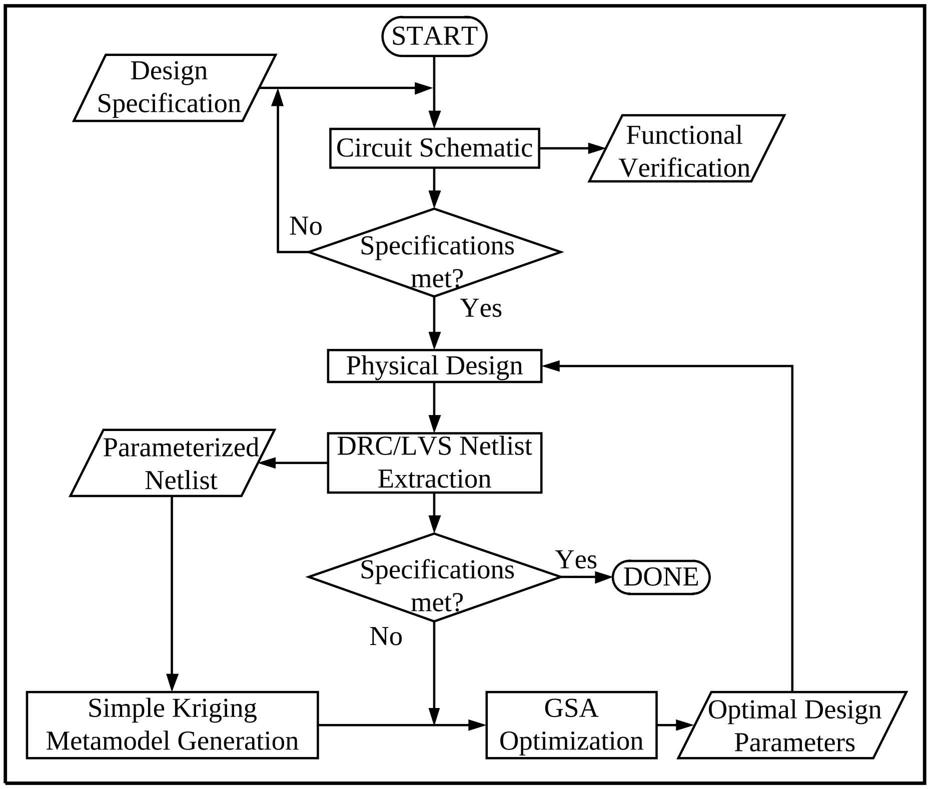


Fig. 2: The proposed design optimization flow.

V. Method for Simple Kriging Metamodel Generation

The proposed method for the simple kriging metamodel generation is summarized in Fig. 3. This process takes in as input the parameterized parasitic netlist and the design parameters. The first step involves obtaining sample points from the multidimensional design space from which the simple Kriging metamodel is generated. Sample performance points are produced from SPICE simulations using the LHS generated points. These points are fed into the Kriging metamodel generator along with the design points to be estimated. After the metamodel function points are generated, the model's accuracy is analyzed by using random functional objective points for statistical analysis. To test the validity of the generated metamodel, random test points are also simulated with SPICE and used for accuracy analysis using the Mean Square Error (MSE), the Root Mean Square Error (RMSE) and Correlation Coefficient (R^2) metrics. The statistically accurate metamodel is obtained as the output, in this case a simple Kriging model.

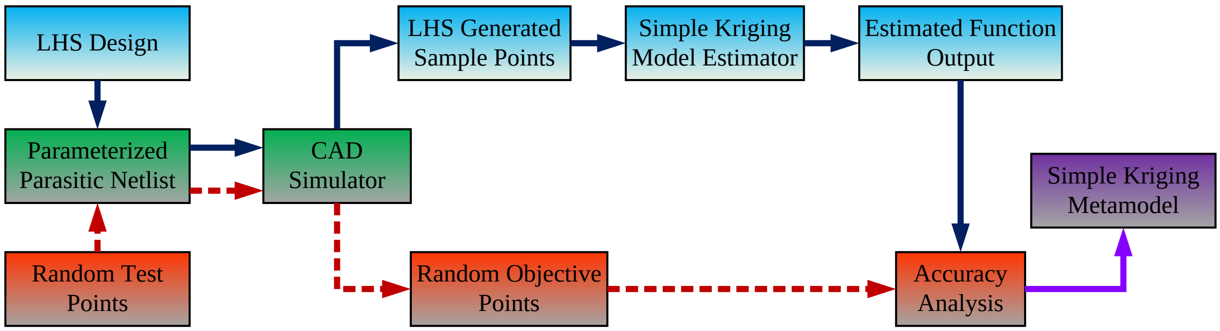


Fig. 3: The proposed simple Kriging metamodel generation flow.

The sample points are generated using the LHS method which was chosen because it covers all input dimensions simultaneously and thus improves the variance over completely random techniques. In [4], a comparison of sampling techniques shows that LHS generates more accurate models over random sampling (Monte Carlo). Specifically, the variance in the response $f(\mathbf{x})$ at n LHS sample points is given by the following [24]:

$$\mathrm{Var}(\overline{y}_{LHS}) = \left(\frac{1}{n}\right)\mathrm{Var}(f(\mathbf{x})) - \left(\frac{k}{n}\right) + o\left(\frac{1}{n}\right), \tag{1}$$

where k is a positive constant and is shown to be smaller than the variance of random sampling techniques.

The general expression of a Kriging model has the following form:

$$y(\mathbf{x_0}) = \sum_{j=1}^{L} \lambda_j B_j(\mathbf{x}) + z(\mathbf{x}), \tag{2}$$

where $y(\mathbf{x_0})$ is a stochastic function which predicts the response at the design point $(\mathbf{x_0})$. $\{B_j(\mathbf{x}), j = 1, \cdots, L\}$ is a specific set of L basis functions over the design domain D_N, λ_j are fitting coefficients to be determined and $z(\mathbf{x})$ is a stochastic process with zero mean and is based on a spatial correlation function called the *variogram*:

$$r(\mathbf{s}, \mathbf{t}) = \mathrm{Corr}(z(\mathbf{s}), z(\mathbf{t})). \tag{3}$$

The variogram is used to derive the weights, λ_j. The autocorrelation of the design points is characterized by the covariance function [25]. The weights are chosen so that the Kriging variance is minimized [27, 26].

Common variations of Kriging include simple, ordinary and universal Kriging. In this paper, we use the *simple Kriging method*. It assumes a constant and known mean over the global domain. Assuming that there are n sample points of the input variable x, to predict a new point (response) $y(x_0)$, the weights λ are estimated by the following:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \Gamma^{-1} \begin{pmatrix} \gamma(x_1, x_0) \\ \vdots \\ \gamma(x_n, x_0) \end{pmatrix}. \tag{4}$$

Γ is the covariance matrix:

$$\Gamma = \begin{pmatrix} \gamma(x_1, x_1) & \cdots & \gamma(x_1, x_n) \\ \vdots & \ddots & \vdots \\ \gamma(x_n, x_1) & \cdots & \gamma(x_n, x_n) \end{pmatrix}, \tag{5}$$

where the covariance is calculated by:

$$\gamma(x_1, x_2) = E\left(|z(x_1) - z(x_2)|^2\right). \tag{6}$$

The most common variogram model used is the spherical and is expressed by the following expressions:

$$\gamma(h) = C_0 + C\left(\frac{3h}{2a} - \frac{1}{2}\left(\frac{h}{a}\right)^3\right) \text{ for } 0 < h \leq a, \tag{7}$$

where C_0, C and a are shape parameters.

The Kriging prediction function is generated with the use of the MATLAB toolbox mGstat [28]. The input is the set of design variables, the points to be estimated and Kriging parameters to select the Kriging method. The generated metamodels must be validated before use for design exploration or optimization. Validation tests ensure the accuracy of the metamodel and are usually done with additional random points through statistical analysis. The metrics Mean Square Error (MSE), Root Mean Square Error (RMSE) and the Correlation Coefficient R^2 are used. A lower value for both MSE and RMSE and an R^2 value close to unity imply a more accurate model.

VI. Proposed Gravitational Search Algorithm

The gravitational search algorithm (GSA) was introduced in 2009 [5] as a new heuristic optimization algorithm based on the Newtonian laws of gravity. The algorithm models the search agents as mass objects. Search agents who perform better by finding more quality solutions (agents in locations of design points with superior performance) are designated heavier masses while those with poor solutions have lighter masses. The interactions of the search agents with each other are developed using the principle of Newtonian laws; the heavier mass agents exert a much greater force and attract other agents with smaller masses (smaller attractive forces), hence pulling search agents with previous poor solutions towards areas with likely optimal solutions. With this technique agents with heavier masses move slower and explore more of the optimal solution area, while lesser mass agents move faster without concentrating on design space areas with poor solutions. This principle implies two useful features: design space exploration and exploitation. The exploration feature is the capability of the algorithm to actively stratify the design space while exploitation is the efficiency of locating optimal solutions in a likely optimal area.

A high-level overview of the GSA algorithm is shown in Fig. 4. The search agents, for example a set of design parameters, are denoted by their locations and masses as M_w, M_x, M_y, and M_z in a design space. The location of each agent at any particular time is shown and the quality of solution is denoted by the mass size of the agent. M_z, currently has the best quality while M_w has the worst. The underlying principle of the algorithm is shown using the forces acting on search agent M_y as an example.

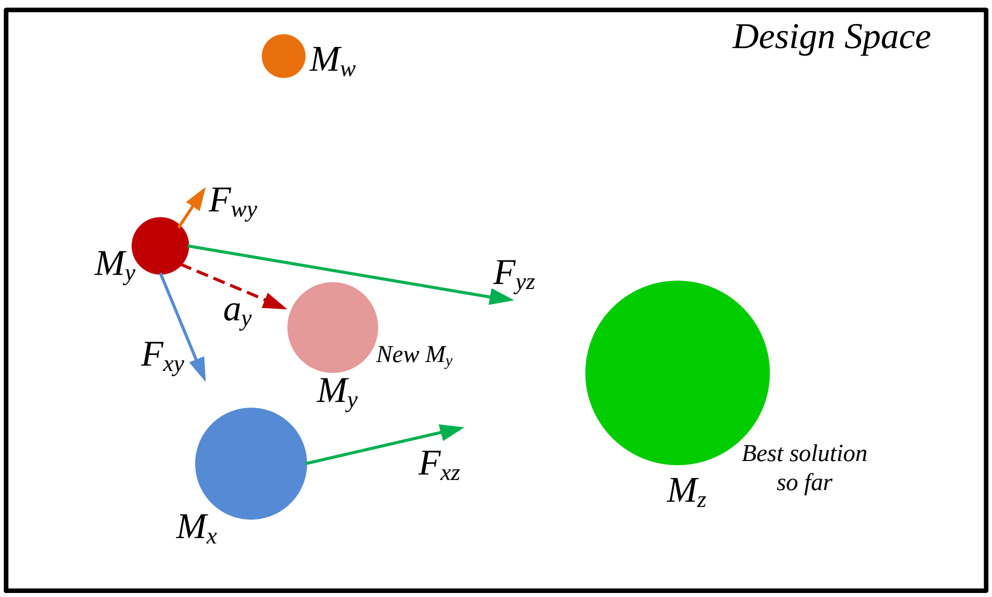


Fig. 4: GSA: Search agents are attracted towards locations with possible quality solutions

Assume a system with N denoting the number of masses (search agents/nodes). The location (design point) of the ith mass can be expressed in functional form as follows:

$$X_i = (x_i^1, x_i^2, \ldots, x_i^d, \ldots, x_i^n) \text{ for } i = 1, 2, \ldots, N, \tag{8}$$

where x_i^d, presents the position of the ith agent in the dth dimension, and n is the number of dimensions.

In calculating the force of attraction, a random value has been added to provide a stochastic element to the algorithm and to reduce the likelihood of optimization being stuck in local minima. The attractive force on a mass object 'i' from a mass object 'j' is given by:

$$F_{ij}^d(t) = G(t)\left(\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon}\right)\left(x_j^d(t) - x_i^d(t)\right), \tag{9}$$

where M_{aj} and M_{pi} are the active and passive gravitational masses of objects 'j' and 'i' respectively, $G(t)$ is a gravitational constant at time t, and R_{ij} is the Euclidean distance between the two objects. The mass of each agent is updated with the following expressions:

$$M_i(t) = \left(\frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}\right), \tag{10}$$

$$m_i(t) = \left(\frac{fit_i(t) - worst(t)}{best(t) - worst(t)}\right), \tag{11}$$

where $fit_i(t)$ represents the best solution found in each iteration. Thus, the total force acting on an object is expressed as:

$$F_i^d(t) = \sum_{j=1.j\neq i}^{N} rand_j F_{ij}^d(t), \tag{12}$$

where $rand_j$ is a random number between 0 and 1.

The velocity update is calculated by the following expression:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t), \tag{13}$$

where $rand_i$ is also a random number between 0 and 1 and $a_i^d(t)$ is the total force in Eqn. 9 divided by the mass of the object in Eqn. 10. The new agent location update is given by:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{14}$$

A convergence of the masses by the heaviest mass presents an optimal solution. One appealing feature of the GSA is that it is memoryless, as it does not need to remember previous best solutions but still guarantees a near-optimal solution by virtue of mass acquisition. These qualities of the GSA naturally lend themselves to design optimization problems of AMS circuits.

GSA implemented for the thermal sensor design is shown as pseudocode in Algorithm 1. The algorithm takes as input the design objective and design parameters along with the validated simple Kriging metamodel. Its outputs are the optimal design parameters points (converged search locations). In the pseudocode, steps 1 – 3 initialize the optimization flow by setting up the maximum number of iterations and the number of mass agents to use. Step 4 sets up the location of each of the search nodes with generic masses. Steps 7 – 14 consist of the main section which analyzes each search node per iteration and updates the mass, velocity and location, reiteratively until an optimal solution is found or the termination criterion is met.

Algorithm 1: GRAVITATIONAL SEARCH ALGORITHM FOR OPTIMIZATION OVER METAMODELS.

Input: Optimization design objective and design variables with parameterized netlist

Output: Optimal design parameters for design objective

Initialize iteration counter: $counter \leftarrow 0$;

Initialize maximum number of iterations Max_{iter};

Initialize number of search agents η, gravity constant G, and velocity ν;

Consider the objective of interest Power_{TS_i};

$i \leftarrow 2$;

while *($counter < Max_{iter}$)* **do**

  Evaluate objective of interest (Power_{TS_i}) for each search node.;

  Update best and worst solution per function objective.;

  Update the gravity constant G.;

  Calculate M and a for each search node.;

  Update ν for each search node.;

  Update search nodes by applying velocity on M.;

  $counter \leftarrow counter + 1$.;

end

return *location*;

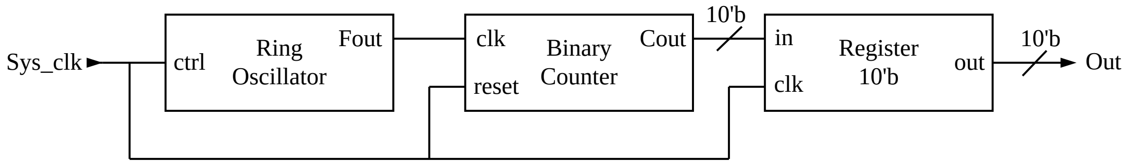
VII. EXPERIMENTAL RESULTS

The tools used for the design are the schematic and layout editors of the Cadence Virtuouso platform on a 45 nm process design kit. MATLAB was used to implement the metamodel generation and optimization algorithm using the toolboxes, mGstat [28], and GSA [5].

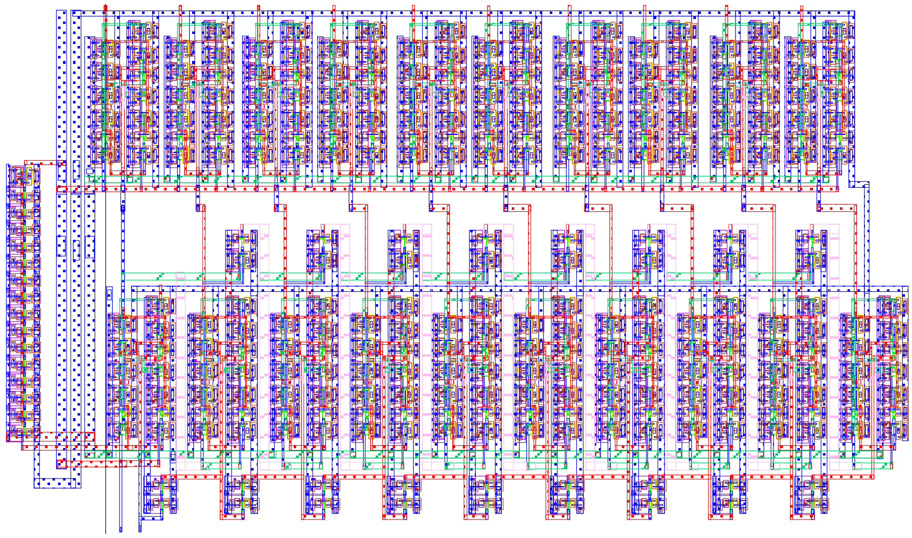
A. Case Study Circuit: a 45 nm Thermal Sensor

The design proposed in [20] serves as the basis for the thermal sensor used as a case study in this work. The design presented here is implemented with the conventional ring oscillator topology and is not operated in the subthreshold region. The system-level block diagram, which consists of three 3 major components, is shown in Fig. 5a [20].

The 10-bit binary counter consists of JK flip-flops. The 10-bit register stores the value from the counter and is also implemented with JK flip-flops. The ring oscillator consists of a cascade of 15 inverters



(a) Block diagram.



(b) Physical design.

Fig. 5: Design of the 45 nm CMOS thermal sensor.

connected in a loop. The oscillation frequency is expressed as:

$$f_{osc} = \frac{1}{n(t_{pLH} + t_{pHL})}, \tag{15}$$

where n is the number of stages used in the oscillator and t_{pLH} and t_{pHL} are the low-to-high and high-to-low propagation delays, respectively. Ideally, the propagation delays can be expressed as [22]:

$$t_{pLH} = \left(\frac{-2C_L V_{tp}}{\kappa_p(V_{dd}+V_{tp})^2} + \frac{C_L}{\kappa_p(V_{dd}+V_{tp})} \ln\frac{1.5V_{dd}+2V_{tp}}{0.5V_{dd}}\right), \tag{16}$$

$$t_{pHL} = \left(\frac{2C_L V_{tn}}{\kappa_n(V_{dd}-V_{tn})^2} + \frac{C_L}{\kappa_n(V_{dd}-V_{tn})} \ln\frac{1.5V_{dd}-2V_{tn}}{0.5V_{dd}}\right), \tag{17}$$

where C_L is the capacitive load and κ_n and κ_p are the transconductance values given by:

$$\kappa_{n/p} = \mu_n C_{ox}\left(\frac{W}{L}\right)_{n/p}. \tag{18}$$

In Eqn. 15 - Eqn. 17, the threshold voltage V_t and mobility μ are most sensitive to temperature and are given by [23]:

$$V_t(T) = V_t(T_0) + \alpha_{V_t}(T - T_0), \tag{19}$$

$$\mu(T) = \mu_0 \left(\frac{T}{T_0}\right)^{\alpha_\mu}, \tag{20}$$

where, $\alpha_{V_t} = -0.5 - 3.0mV/^\circ K$ and $\alpha_\mu = -1.2 - 2.0$. An increase in temperature leads to an increase in the propagation delay which results in a decrease of the oscillating frequency.

The technology library used for the implementation of this thermal sensor is a 45 nm process design kit provided by Cadence. The thermal sensor design is characterized to sense temperatures between 0°C and 100°C. The *Sys_clk* signal is used to enable the thermal sensor. When the *Sys_clk* turns to logic zero, the ring oscillator is disabled, the counter is also reset and the register also stops saving the count, storing the last count value it had before the *Sys_clk* was set to logic "0". The binary counter is used to count the frequency difference between the ring oscillator output and the system clock. The count is stored in the 10-bit register and calibrated to measure the temperature change. The physical design of the thermal sensor is shown in Fig. 5b.

The performance and accuracy of the physical design of the thermal sensor is degraded when compared to the schematic design. A comparison is presented in Table I. The power consumption is increased by 29%. This circuit exhibits a linear dependence of oscillation frequency on junction temperature as shown in Fig. 6.

TABLE I: Thermal sensor characteristic for the baseline design.

Design	Average Power (P_{TS})	Sensitivity, (T_{TS})	Area (μm^2)
Schematic	293.1 μW	16.88 MHz/°C	-
Layout	379.4 μW	9.42 MHz/°C	1221.37
% Change	29.44%	44.2%	-

The frequency response of the schematic design is 5.924 GHz (at 0°C) to 4.236 GHz (at 100°C). Assuming a 6 GHz maximum clock rate for the ring oscillator, and a 10 bit counter (1024 maximum count) the effective resolution is 0.097°C/bit resolution. The range of frequency output is severely degraded by parasitics, as seen in Fig. 6. The range drops to 3.867 GHz (0°C) and 2.986 GHz (100°C). There is a 47.8% change in frequency/temperature resolution by comparing the schematic design to the physical design. The area of the physical design is 1221.37 μm^2.

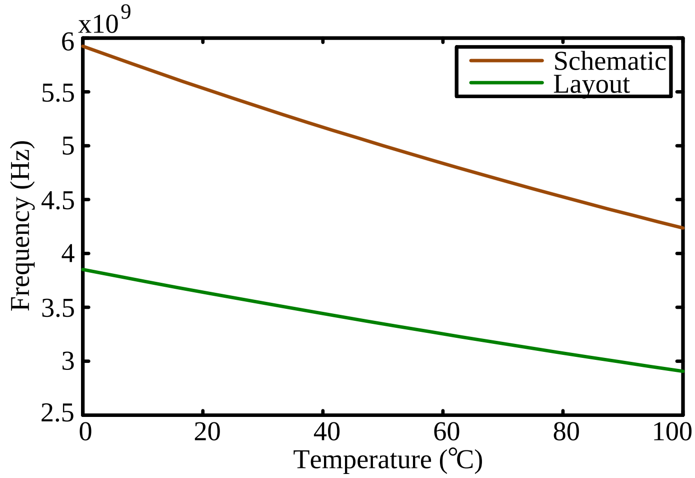


Fig. 6: Frequency response versus temperature for the thermal sensor.

B. Results Analysis

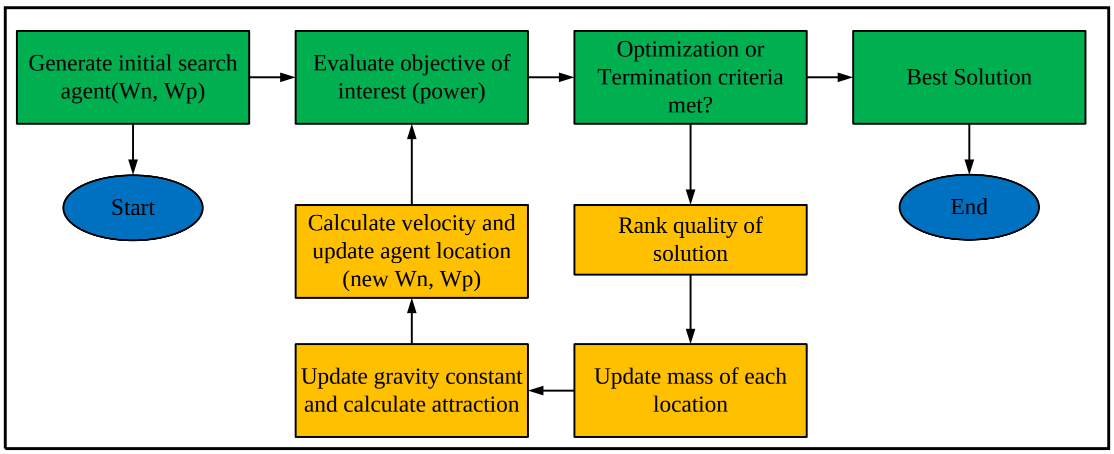
For this case study, *six design parameters were chosen*, based on the 3 components of the thermal sensor. The widths of the NMOS and PMOS transistors in the RO are parameterized to WN_{osc} and WP_{osc}, respectively. The widths of the transistors for the 10-bit counter and 10-bit registers are parameterized to WN_{ctr}, WP_{ctr}, WN_{reg} and WP_{reg}, respectively. In generating the Kriging metamodels, 100 sample points were obtained from LHS. To evaluate the accuracy of the generated metamodel, the metrics discussed in Section V are used as shown in Table II.

TABLE II: Accuracy analysis of the simple Kriging metamodels.

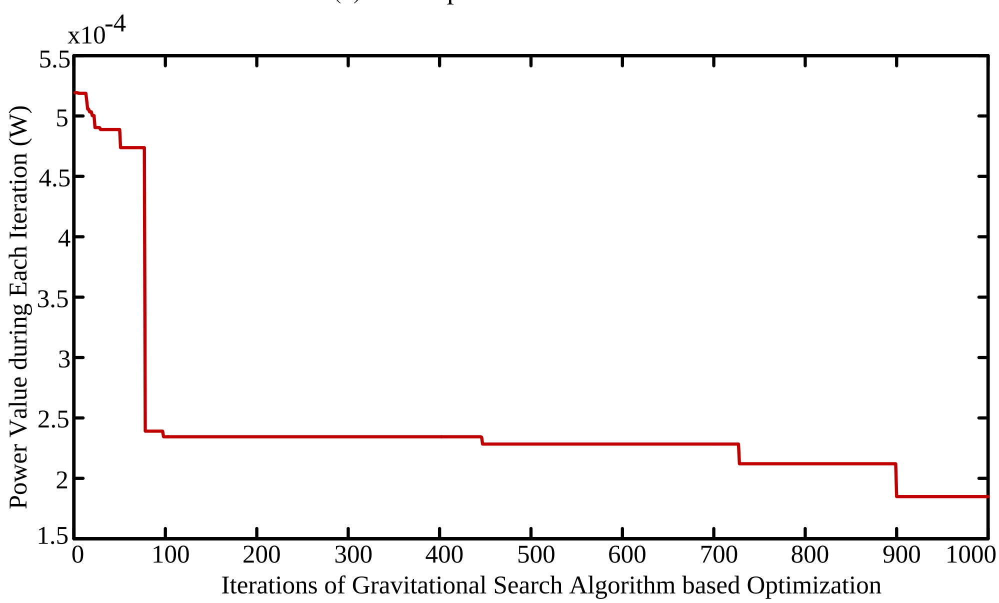
Metric	Value
Mean Square Error (MSE)	4.36×10^{-18}
Root Mean Square Error ($RMSE$)	2.09×10^{-09}
Coefficient of Determination (R^2)	0.9934

From the results in Table II, the Kriging metamodels are sufficiently accurate with very low MSE and RMSE values. The correlation coefficient R^2 is very close to unity. The total time taken for the metamodel generation was approximately 30 hours, *the bulk of this time being the simulation time required for the sample points*. The time however is a factor of 10 lower than the approximately 300 hours required for an exhaustive simulation of the design across the entire design space.

In optimizing the thermal sensor, the GSA optimization is applied to the generated metamodel with an initial number of 50 search agents and a maximum iteration of 1000 runs. The design objective is the minimization of power consumption. The optimization flow is shown in Fig. 7a and the results are shown in Fig. 7b. From the optimization graph, it is seen that the algorithm is able to reach an optimized solution of 184.7 μW in about 900 iterations. The gravitational search being a heuristic algorithms its convergence is usually near optimal. We chose 1000 iterations based on results from previous experiments. In the implementation of the algorithm, there is also a termination criterion where the algorithm could also terminate before the maximum iteration.



(a) GSA optimization flow.



(b) GSA performance on Kriging metamodel for the 45 nm thermal sensor.

Fig. 7: GSA optimization.

The final design parameters are shown in Table III and the optimized responses of the thermal sensor are provided in Table IV. Compared to the schematic baseline design, there is a 36.9% reduction in power dissipation with an area penalty of about 45%.

TABLE III: Final Design Parameters obtained from the Kriging metamodel optimization.

Design Parameter	Final Size Values
WN_{osc}	215 nm
WP_{osc}	140 nm
WN_{ctr}	313 nm
WP_{ctr}	121 nm
WN_{reg}	224 nm
WP_{reg}	378 nm

TABLE IV: Thermal Sensor Output Comparison

Design	Average Power, (P_{TS})	Sensitivity, (T_{TS})	Area (μm^2)
Schematic	293.1 μW	16.88 MHz/°C	-
Layout	379.4 μW	9.42 MHz/°C	1221.37
Final	184.7 μW	9.42 MHz/°C	1770.98*
% Change	36.9%	44.2%	45%*

C. Comparative Perspective with Related Research

The scope of this paper is fast analog design optimization using geostatistical based methods. While a thermal sensor has been used as a case study circuit the proposed methodology is very much applicable for optimization of any other analog design. Many thermal sensor designs have been presented in the literature including [20, 19, 23]. However, the technology node, operating voltage, topology, and design objective are quite different and hence a fair comparison is not possible. The proposed thermal sensor design has an improved sensitivity of 0.097°Cwhich is higher than the other selected designs. The overall power consumption is 184.7 μW which is higher than the design presented in [20]. The design in [20], however has an operating voltage of 0.3 V compared to an operating voltage of 1 V for our design. Our design also has a smaller area overhead cost of 0.001 mm^2 compared to the related designs. The 45 nm technology is similar to the thermal sensor presented in [19] which also had a low area of 0.04

mm^2. A perspective comparison of the performance of existing techniques is shown in Table V. The ring oscillator design in [29] implements Tabu Search (TSA) and Simulated Annealing (SAA), and the performance results are compared with the thermal sensor in this paper. The RO has 6 transistors and 2 design parameters with the TSA and SSA running for 12 and 15 iterations, while the thermal sensor has 896 transistors and 6 parameter designs and optimizing in about 900 iterations. The computational time in Table V has been normalized to compare with the thermal sensor design. The proposed method has improved simulation and optimization times of 17.46 sec. as compared to 241.25 sec. and 31.04 sec. for the TSA and SAA, respectively.

TABLE V: Comparative Analysis of Metamodel and Optimization

Metric	On Netlist			With Metamodel		
	TSA	SSA	GSA	TSA Polynomial	SSA Polynomial	GSA Simple Kriging
Computational Time	4.72 x 10^6 s	4.73 x 10^6 s	1.08 x 10^6 s	241.25 s	31.04 s	17.46 s

VIII. Conclusions

In this paper, a new design optimization flow incorporating a geostatistical inspired metamodeling technique (simple Kriging) and a gravitational search algorithm for analog/mixed signal circuit and system design optimization has been presented. The proposed methodology has been illustrated with the design optimization of a 45 nm CMOS based thermal sensor. Simple Kriging based metamodeling produces very accurate metamodels while reducing the time for exhaustive exploration of the design space by approximately 90%. A total of six design parameters were considered for metamodeling and optimization. The gravitational search algorithm also optimizes the design by reducing the power consumption by 36.9%. In future research, the metamodeling technique will be extended for process variation effects and statistical optimization. The proposed methodology will also be extended for multi-objective optimization schemes.

Acknowledgments

This research is supported in part by NSF award DUE-0942629. This archival journal paper is based on our conference publication [30]. The authors would like to thank Dr. Oleg Garitselov for his help.

REFERENCES

[1] Biles, W. E., Kleijnen, J. P. C., van Beers, W. C. M., and van Nieuwenhuyse, I.: 'Kriging Metamodeling in Constrained Simulation Optimization: An Explorative Study', Proc. Winter Simulation Conf., Dec. 2007, pp. 355–362

[2] Ankenman, B., Nelson, B., and Staum, J.: 'Stochastic Kriging for Simulation Metamodeling', Proc. Winter Simulation Conf., Dec. 2008, pp. 362–370

[3] Dellino, G., Kleijnen J., and Meloni, C.: 'Robust Simulation-Optimization using Metamodels', Proc. Winter Simulation Conf., Dec. 2009, pp. 540–550

[4] Garitselov, O., Mohanty, S., and Kougianos, E.: 'A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits', IEEE Trans. Semiconductor Manufacturing, 2012, 25, (1), pp. 26–36

[5] Rashedi, E., Nezamabadi-pour, H.and Saryazdi, S.: 'GSA: A Gravitational Search Algorithm', Information Sciences, 2009, 179, (13), pp. 2232–2248

[6] Binder, T., Heitzinger, C., and Selberherr, S.: 'A Study on Global and Local Optimization Techniques for TCAD Analysis Tasks', IEEE Trans. Computer-Aided Design of Integrated Circuits and Syst., 2004, 23, (6), pp. 814–822

[7] Aggarwal, V.: 'Analog Circuit Optimization using Evolutionary Algorithms and Convex Optimization'. Master's thesis, Massachusetts Institute of Technology, 2007.

[8] Lim, D., Ong, Y. -S., Jin, Y., and Sendhoff, B.: 'A Study on Metamodeling Techniques, Ensembles, and Multi-Surrogates in Evolutionary Computation', Proc. Conf. on Genetic and Evolutionary Computation, New York, USA, 2007, pp. 1288–1295

[9] Wang, G. G., and Shan,S.: 'Review of Metamodeling Techniques in Support of Engineering Design Optimization', Journal of Mechanical Design, 129, 2007, (4), pp. 370–380

[10] Khosravi, A., Nahavandi, S., and Creighton, D.: 'Developing Optimal Neural Network Metamodels Based on Prediction Intervals', Int. Joint Conf. Neural Networks, June 2009, pp. 1583–1589

[11] You, H., Yang, M., Wang, D. and Jia, X.: 'Kriging Model Combined with Latin Hypercube Sampling for Surrogate Modeling of Analog Integrated Circuit Performance', Proc. Int. Symp. Quality of Electronic Design, 2009, pp. 554–558

[12] Yu, G., and Li, P.: 'Yield-Aware Analog Integrated Circuit Optimization Using Geostatistics Motivated Performance Modeling', IEEE/ACM Int. Conf. Computer-Aided Design, Nov. 2007, pp. 464–469

[13] Okobiah, O., Mohanty, S. P., Kougianos, E., and Garitselov,O.: 'Kriging-Assisted Ultra-Fast Simulated-Annealing Optimization of a Clamped Bitline Sense Amplifier', Int. Conf. VLSI Design, Jan. 2012, pp. 310–315

[14] Okobiah, O., Mohanty, S. P., Kougianos, E.: 'Ordinary Kriging Metamodel-Assisted Ant Colony Algorithm for Fast Analog Design Optimization', Int. Symp. Quality Electronic Design, Mar. 2012, pp. 458–463

[15] Li, Y. and Li, J: 'Swarm Intelligence Optimization Algorithm Based on Orthogonal Optimization', Int. Conf. Computer Modeling and Simulation, January 2010, pp. 12–16

[16] Garitselov, O., Mohanty, S. P., and Kougianos, E.: 'Accurate Polynomial Metamodeling-Based Ultra-Fast Bee Colony Optimization of a Nano-CMOS Phase-Locked Loop', J. Low Power Electronics, 2012, 8, (3), pp. 317–328

[17] Cashero, Z. Chen, A. Hoppal, R. and Chen, T.: 'Fast Evaluation of Analog Circuits Using Linear Programming', IEEE Symp. Computer Society, July 2010, pp. 253–258

[18] Kashfi, F., Hatami, S., and Pedram, M., 'Multi-Objective Optimization Techniques for VLSI Circuits', Proc. Int. Symp. Quality of Electronic Design, March 2011, pp. 1–8

[19] Datta, B., and Burleson, W.: 'Low-Power and Robust On-Chip Thermal Sensing Using Differential Ring Oscillators', Midwest Symp. Circuits and Systems, Aug. 2007, pp. 29–32

[20] Park, S., Min, C., and Cho, S.-H. ‘A 95nW Ring Oscillator-based Temperature Sensor for RFID Tags in 0.13 μm CMOS’, IEEE Int. Symp. Circuits and Syst., May 2009, pp. 1153–1156.

[21] Zhang, Y., and Srivastava, A.: ‘Accurate Temperature Estimation Using Noisy Thermal Sensors’, ACM/IEEE Design Automation Conf., July 2009, pp. 472–477

[22] Meng, T., and Xu, C.: ‘A Cross-Coupled-Structure-Based Temperature Sensor with Reduced Process Variation Sensitivity’, J. Semiconductors, 2009, 30, (4), pp. 1642– 1648

[23] Chen,P. Chen, C. -C., Tsai, C. -C., and Lu, W. -F.: ‘A Time-to-Digital-Converter-Based CMOS Smart Temperature Sensor,” IEEE Journal of Solid-State Circuits, 2005, 40, (8), pp. 1642–1648

[24] Owen, A. B.: ‘A Central Limit Theorem for Latin Hypercube Sampling,” Journal of the Royal Statistical Society, Series B, 1992, 54, pp. 541–551

[25] Bohling, G.: ‘Kriging’, Kansas Geological Survey, Tech. Rep., 2005

[26] Cressie, N. A. C.: Statistics for Spatial Data (Wiley, 1993)

[27] van Beers, W.: ‘Kriging Metamodeling in Discrete-Event Simulation: An Overview”, Proc. Winter Simulation Conf., 2005, pp. 202–208

[28] mGstat: A Geostatistical Matlab Toolbox. [Online]. Available: mgstat.sourcefourge.net

[29] Garitselov, O. Mohanty, S. P., and Kougianos, E.: ‘Fast Optimization of Nano-CMOS Mixed-Signal Circuits Through Accurate Metamodeling’, Int. Symp. Quality Electronic Design, Mar 2011, pp. 405–410,

[30] Okobiah, O., Mohanty, S. P., and Kougianos, E.: ‘Geostatistical-Inspired Metamodeling and Optimization of Nano-CMOS Circuits’, Proc. IEEE Computer Soc. Annual Symp. VLSI, Aug. 2012, pp. 326–331.