Enhanced Metamodeling Techniques for High-Dimensional IC Design Estimation Problems

Andrew B. Kahng[+ †], Bill Lin[†] and Siddhartha Nath[+]
[+]CSE and [†]ECE Departments, University of California at San Diego.
abk@ucsd.edu, billlin@ucsd.edu, sinath@ucsd.com

Abstract*—Accurate estimators of key design metrics (power, area, delay, etc.) are increasingly required to achieve IC cost reductions in system-level through physical layout optimizations. At the same time, identifying physical or analytical models of design metrics has become very challenging due to interactions among many parameters that span technology, architecture and implementation. *Metamodeling* techniques can simplify this problem by deriving *surrogate models* from samples of actual implementation data. However, the use of metamodeling techniques in IC design estimation is still in its infancy, and practitioners need more systematic understanding. In this work, we study the accuracy of metamodeling techniques across several axes: (1) low- and high-dimensional estimation problems, (2) sampling strategies, (3) sample sizes, and (4) accuracy metrics. To help obtain more general conclusions, we study these axes for three very distinct chip design estimation problems: (1) area and power of networks-on-chip routers, (2) delay and output slew of standard cells under power delivery network noise, and (3) wirelength and buffer area of clock trees. Our results show that (1) adaptive sampling can effectively reduce the sample size required to derive surrogate models by up to 64% (or, increase estimation accuracy by up to 77%) compared with Latin hypercube sampling; (2) for low-dimensional problems, Gaussian process-based models can be 1.5x more accurate than tree-based models, whereas for high-dimensional problems, tree-based models can be up to 6x more accurate than Gaussian process-based models; and (3) a variant of weighted surrogate modeling [7], which we call *hybrid surrogate modeling*, can improve estimation accuracy by up to 3x. Finally, to aid architects, design teams, and CAD developers in selection of the appropriate metamodeling techniques, we propose guidelines based on the insights gained from our studies.

I. Introduction

In advanced technology nodes, IC product design faces tremendous challenges of power and cost reduction even as performance and utility continue to scale [22]. Effective design space exploration and design optimization increasingly depend on the accurate modeling and estimation of key design metrics (e.g., power, area or delay) throughout the system-level through physical layout flow. Such modeling and estimation tasks are complicated by the fact that the relevant design metrics are affected by a tremendous number and range of parameters - microarchitectural, design implementation, operational, technology, and manufacturing. Moreover, the space of possible design outcomes grows combinatorially with the number of parameters, so estimation models must be obtained with relatively sparser numbers of data points.

In general, derivation of physical or analytical models is very difficult because at high dimensions, complex interactions between parameters are hard to predict, and at low dimensions, artifacts of the optimizations within IC design tools are hard to model accurately [13]. Recent works (e.g., [12], [19]) have shown that *metamodeling* techniques can be effective in creating *surrogate models* from sample data that is obtained from the actual chip implementation tools or flows. Moreover, metamodeling techniques have been demonstrated to have high accuracy and low modeling overhead for several IC design estimation applications. However, as there has yet been only limited use of metamodeling techniques for IC design estimation, practitioners lack systematic understanding and guidance regarding how to choose and apply the available techniques.

In this work, we study the accuracy of three popular metamodeling techniques [9], Multivariate Adaptive Regression Splines (MARS), Radial Basis Functions (RBF), and Kriging (KG), for both low- and high-dimensional modeling applications. We also study *hybrid surrogate modeling* (HSM), a variant of [7], which obtains improved estimates by finding weighted combinations of estimates from individual surrogate models. We furthermore study two sampling strategies, Latin Hypercube Sampling [11] and Adaptive Sampling [6], to understand the impacts of training set methodology on modeling efficiency and accuracy. Finally, we also study the impact of the number of dimensions (D) and the sample size (N) on accuracy. Overall, the axes of our study are

- quality-of-results metrics: maximum and average errors;
- resource metrics: number of dimensions (D), number of samples (N);
- modeling techniques: MARS, RBF, KG, and HSM; and
- sampling strategies: LHS, AS.

To help ensure generality of our observations, we study the above axes for three distinct types of estimation problems in IC design.

- *NoC*: estimation of *area* and *power* of Network-on-Chip (NoC) routers across microarchitectural and implementation parameters. Following the approach of [13], we formulate this as a *low-dimensional* (low-D) modeling problem based on microarchitectural (e.g., flit-width, number of virtual channels) and implementation (e.g., clock frequency) parameters.
- *PDN*: estimation of *delay* and *output slew* of a standard cell (inverter) in the presence of power delivery network (PDN) noise. Extending the work of, e.g., [4], we formulate this as a *high-dimensional* (high-D) modeling problem based on implementation (e.g., load capacitance, offset of voltage noise) and technology (e.g., threshold voltage, process corner) parameters.
- *CTS*: estimation of *wirelength* and *buffer area* of clock trees obtained from a commercial clock tree synthesis (CTS) flow. We formulate this as a high-D modeling problem based on implementation (e.g., skew bound, max transition time) and technology (e.g., wire widths and parasitics) parameters.

Our results reported below offer a number of practically valuable insights. For example, we observe that RBF and KG are more accurate than MARS for low-D problems such as NoC, whereas MARS is significantly more accurate than RBF and KG for high-D problems such as PDN and CTS. The HSM technique can be up to 3x more accurate than the best surrogate model across low- and high-D problems. We also observe that AS is always superior to LHS, e.g., AS reduces the simulation (sample generation) overhead by up to 64% for a given model accuracy requirement, and reduces worst-case estimation errors by up to 77% for a given sampling budget. Our experiments also shed light on threshold and tradeoffs related to issues such as the "curse of dimensionality" and risk of overfitting. The key contributions of our work are summarized as follows.

- We demonstrate the accuracy limits of metamodeling techniques for low- and high-D modeling in IC design applications across multiple axes. We show that at low-D, RBF and KG can be up to 1.5x more accurate than MARS, whereas at high-D MARS can be up to 6x more accurate than RBF and KG.
- We achieve reduced sample (training set) generation overheads by using the latest AS techniques [6]. AS achieves up to 77% reduction in worst-case estimation error or up to 64% reduction in N as compared to LHS with equivalent resources.
- We apply HSM, a type of weighted surrogate modeling [7], to surrogate models generated by MARS, RBF, and KG by adding weights to the individual estimates. HSM achieves up to 3x reduction in the worst-case estimation error.

978-3-9815370-0-0/DATE13/©2013 EDAA

- We provide high-level metamodeling guidelines for estimation problems in IC design to aid architects, design teams, and CAD developers.

In the remainder of this paper, Section II reviews previous works, and Section III presents background on metamodeling techniques, multicollinearity, and measures of metamodeling accuracy in high dimensional applications. Section IV describes our methodologies for three types of modeling problems in IC design, namely, power and area estimation for NoCs, standard-cell delay and slew estimation under PDN noise, and wirelength and buffer area estimation for CTS. Section V presents results on estimation errors across all axes of our study for each type of problem. In Section VI, we summarize our work and outline directions for future work.

II. Related Work

Surrogate models are gaining popularity for early exploration and characterization of the solution space for various aspects of IC design. These models are broadly classified as (1) Gaussian process-based models (GPM), and (2) tree-based models [9]. MARS is an additive tree-based model, whereas RBF and KG are GPMs. Kahng et al. [12] use MARS with linear splines to estimate NoC area and power. They report about 60% worst-case and about 6% average errors for both area and power. Cheng et al. [4] also report similar errors in modeling the worst-case performance under power supply noise variation by using MARS. Ilumoka [10] uses RBF to estimate interconnect crosstalk but does not quantify the estimation errors as compared to SPICE simulations. Liu [15] uses KG to model IR drop and on-chip temperature and reports worst-case error to be within 76%. Goel et al. [7] propose weighted surrogate modeling instead of using the best estimation model. Our study of HSM below applies the idea of [7] using least-squares regression to determine the weights.

Sample selection is a key step in obtaining accurate and robust surrogate models. LHS [11] is a popular sampling technique that selects samples uniformly across the ranges of the input parameters. However, it is not effective (1) when systems have complex interactions such that the response changes non-monotonically with certain input combinations, while changing uniformly with other input combinations [6], or (2) when D increases, which can exponentially increase the number of samples required to generate accurate surrogate models. AS is gaining popularity as it uses fewer samples as compared to LHS to derive surrogate models that are highly accurate. Gorissen et al. [8] use AS to study accuracy, as well as impacts of D and N, in various surrogate models for low-noise amplifier (LNA) gain estimation. Although they quantify the root mean square error (RMSE) in each experiment, it is difficult to assess when some metamodels will perform better than others from the characteristics of the input parameters; thus, they propose to try all models and choose the best one. Zhu et al. [20] use AS to demonstrate significant reduction in size of lookup tables to store PVT-sensitive I-V characteristics. They report average error of 6%, but do not mention the worst-case error.

III. Metamodeling Background

We apply metamodeling techniques[1], MARS, RBF, and KG, to three types of problems in IC design. At a high level, these techniques model the response as a weighted sum of functions of input parameters plus noise. Formally, suppose we have N values of D parameters for which we know the response, we denote a vector of parameters as $\vec{x} = < x_1, x_2, ..., x_D >$, and its response as $y(\vec{x})$. The set of N values of D parameters and their responses is called a *training set*, each value and its corresponding response is referred to as a *sample*. We derive surrogate models from the training set to predict future response as

$$\hat{y}(\vec{x}) = f(\beta, \vec{x}) + \varepsilon(\vec{x}) \tag{1}$$

where $\hat{y}(\vec{x})$ is the predicted response, $f(\beta, \vec{x})$ expresses the deterministic part of the response, and $\varepsilon(\vec{x})$ is a random noise function. $f(\beta, \vec{x})$ is a linear combination of D known functions and is a realization of the regression model given by

$$f(\beta, \vec{x}) = \beta_0 + \beta_1 f(x_1) + \beta_2 f(x_2) + \cdots + \beta_D f(x_D) \tag{2}$$

where $\beta_0, \beta_1, ..., \beta_D$ are the regression coefficients determined by the regression function, $f(x_i)$. β_0 is typically set to 1 by most models. The metamodeling techniques use different forms of the regression function, $f(x_i)$, where $i = 1, 2, ..., D$, and the random noise function, $\varepsilon(\vec{x})$, to minimize the estimation errors.

Surrogate models are more accurate than ordinary linear fit using *least-squares regression* (LSQR) because they are generalizable, that is, they minimize errors on future input. This is achieved by minimizing the *generalized cross validation* (GCV) error instead of minimizing only the sum of squared errors in the training set. GCV is a mean squared error of the samples in the training set multiplied by a penalty to account for increased variance with increasing model complexity and to prevent overfitting. It is given by

$$GCV = \frac{1}{N} \cdot \left[1 - \frac{C}{N}\right]^{-2} \cdot \sum_{i=1}^{N} [y_i - \hat{y}_i]^2 \tag{3}$$

where y_i and $\hat{y}_i$ denote the response $y(\vec{x})$ and predicted response $\hat{y}(\vec{x})$, respectively, to the i^{th} sample, and C is a penalty to avoid overfitting which is typically equal to D.

The "curse of dimensionality" imposes three important requirements for accurate modeling [19] – (1) more samples are required as D increases, (2) the range of parameters must describe the fitted response surface, and (3) the samples should be independent and identically distributed (iid), that is, they should be minimally correlated. We use two sampling strategies – (1) LHS [11], and (2) AS with *lola-Voronoi* technique [6]. While LHS selects new samples to uniformly cover the ranges of parameters, AS uses a combination of exploitation- and exploration-based methods to select new samples where the residuals are the largest as well as in uncovered ranges of the parameter space.

A. Multivariate Adaptive Regression Splines (MARS)

Multivariate Adaptive Regression Splines are additive tree-based regression methods. The regression function, $f(x_i)$, is a product of spline basis functions which are constructed by using iterative forward and backward steps [9], where basis functions are initially added and pruned later to minimize the GCV error. These steps determine the number of basis functions, number of interactions between parameters, and the *knot*[2] locations. $f(x_i)$ is given by

$$f(x_i) = \prod_{j=1}^{I_i} \left[b_{ji} \cdot (x_v - t_{ji})\right]_+ \tag{4}$$

where I_i is the number of interactions between parameters in the i^{th} basis function, $b_{ji} = \pm 1$, x_v is the v^{th} parameter, t_{ji} is the knot location on each of the corresponding parameters, and the subscript "+" denotes the positive part of a truncated power function.

B. Radial Basis Functions (RBF)

Radial Basis Functions use kernel functions which are symmetric and centered at each parameter in the training set [9]. RBF models in two phases – (1) the *forward selection* phase repeatedly adds kernel functions until the sum of squared errors cannot be minimized, and (2) the *backward selection* phase that prunes these kernel functions such that the GCV error is minimized. The model is represented as

$$f(x_i) = \sum_{j=1}^{N} a_j \cdot K(\mu_j, r_j, x_i) \tag{5}$$

where r_j are scaling factors, $K(.)$ is a kernel function, μ_j are centroids, and a_j are coefficients of the kernel function. Popular choices of kernel functions are Gaussian, cubic, and multiquadrics [27].

C. Kriging (KG)

Kriging is a special kind of interpolation model that models the random noise, $\varepsilon(\vec{x})$, with a weighted correlation model between neighboring values in $\vec{x}$. This is a distinguishing feature of KG because other models treat $\varepsilon(\vec{x})$ as a Gaussian random variable with zero mean and a constant variance [15]. The correlation model is given by

[1] Metamodeling is also referred to as non-parametric regression modeling and surrogate modeling in statistical literature. We use the term "metamodeling" to refer to the modeling techniques and the term "surrogate models" refers to the models derived by these techniques.

[2] Knot is the value of a parameter where a piecewise line segment changes its slope.

$$R(\theta, x_i, x_k) = \prod_{j=1}^{N} R_j(\theta,\ x_k - x_i) \tag{6}$$

where θ is a correlation function parameter. Popular choices of $R_j(\theta,\ x_k - x_i)$ are exponential, Gaussian, linear, spherical, cubic, or spline [16]. Popular choices for the regression function, $f(x_i)$, are linear or quadratic polynomials.

D. Multicollinearity at high dimensions

The vectors of parameters and their responses are also represented as matrices, X, and y, respectively. The dimension of X is $N \times D$, and the dimension of y is $N \times 1$. Matrix $\hat{\beta}$ has estimates for the regression coefficients and is given by

$$\hat{\beta} = \left(X'X\right)^{-1}\left(X'y\right) = R_{xx}^{-1} R_{xy} \tag{7}$$

where R_{xx} is a matrix containing correlations between parameters, x_i and x_j, $(i, j = 1, 2, ..., D)$, and R_{xy} is a matrix containing correlations between $\vec{x}$ and $y(\vec{x})$. The estimate of variance of $\hat{\beta}$ is $\hat{\sigma}^2\left(\frac{1}{N} R_{xx}^{-1}\right)$, where $\hat{\sigma}^2$ is the conditional variance of y given X, and is expressed as [17]

$$\hat{\sigma}^2 = var(y|X) = \left(\frac{N}{N-D}\right) \cdot \frac{|R_{xy}|}{|R_{xx}|} \tag{8}$$

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\hat{\sigma}^2(N-D)}{N} = 1 - \frac{|R_{xy}|}{|R_{xx}|} \tag{9}$$

Multicollinearity arises when parameters are linearly dependent. The matrix X has less than full column rank, is ill-conditioned, and the product $X'X$ is almost singular. The variance of regression coefficients and the coefficient of determination increase according to Equations (8) and (9) when X is ill-conditioned and/or if the responses and vectors of parameters are highly correlated. Various diagnostic tests such as *F-test*, condition number, analysis of variance, and variance inflation factor (VIF) [1], [2], [9] are used to detect ill-conditioned matrices X. In our work, we use the VIF test.

Multicollinearity often occurs in high-D modeling because the probability of linear dependence between parameters increases. Furthermore, it is hard to generate a training set that can avoid this phenomenon. We cannot drop parameters, because for problems in IC design, understanding interactions between parameters may be necessary. For example, given a load (C_ℓ), we need to determine the right sizes (S) of drivers so that delay and slew are minimized. Techniques that use $(X'X)^{-1}$ as a factor (e.g., RBF and KG) may be inaccurate and may result in large estimation errors. *Ridge regression* is a way to "cure" multicollinearity by penalizing large regression coefficients with a factor α [9]. The regression coefficients are estimated by $(X'X + \alpha I)X'y$, where I is the identity matrix.

IV. Modeling Methodologies

We now describe our modeling methodologies for each type of problem – NoC, PDN, and CTS. Table I describes the input parameters and their ranges used to generate the golden data points. The number in parentheses within the "Problem type" column denotes the number of parameters or D of the problem. NoC is an instance of a low-D problem with $D = 5$, whereas PDN and CTS are instances of high-D problems with $D = 11$ and $D = 10$ respectively. The range is represented as $[LB, UB]$, where LB and UB represent lower and upper bounds of parameters.

For the **NoC** problem, we use the *Netmaker* [28] RTL, configure the microarchitectural parameters in the RTL by using values from the ranges described in Table I, and run synthesis, place-and-route (SP&R) simulations by using commercial tools, Synopsys *Design Compiler vF-2011.09-SP4-64* [23] and Cadence *SOC Encounter vEDI10.1* [21]. We extract the total area of standard cells and total power from these simulations. We generate a total of 1024 data points; some of these are used for training and the remainder are used as test data points. We use TSMC65PLUS and TSMC45GS cell libraries in our flows.

For the **PDN** problem, we use voltage parameters to study the complex interactions between these parameters on cell delay and output slew at nominal as well as near-threshold (NTV) regions of operating voltages. Accurate modeling of delay at NTV is still an open problem [14]. Devices operate at NTV when V_{DD} values are close to but slightly larger than V_{TH} values, e.g., 0.5V, 0.6V, etc. We create SPICE netlists of inverters configured with the parameters from Table I and observe delay and output slew for each input combination. To derive surrogate models, we combine V_{DD} and V_{TH} as one parameter $V_{DD} - V_{TH}$, the overdrive voltage, instead of using both V_{DD} and V_{TH}. We generate three million data points by using Synopsys *HSPICE vE-2010.12* [24] simulations and the device models in the TSMC65GPLUS design kit.[3]

TABLE I
Input parameters and their ranges

Problem type	Parameter	Description	Range
NoC	P	# ports	[3, 9]
(5)	V	# VCs	[2, 7]
	B	# buffers	[2, 7]
	F	flit-width	[16, 64]
	C	frequency	[400, 1000]MHz
PDN	S	cell size	x[1, 20]
(11)	C_ℓ	load	[0.9, 84.2]fF
	$Slew_{in}$	input slew	[0.56, 7.09]ps
	N_{amp}	noise amplitude	[0.0, 0.27]V
	N_{slew}	noise slew	[10, 90]ps
	N_{off}	noise offset	[−150, 150]ps
	T	temperature	[233.15, 398.15]K
	V_{DD}	source voltage	[0.5, 1.0]V
	V_{TH}	threshold voltage	[0.14, 0.47]V
	V_{BB}	body bias	[0.05, 0.15]V
	P_{corner}	process corner	[FF, SS, TT]
CTS	M_{sinks}	# sinks	[3, 100]K
(10)	M_{skew}	max. skew	[25, 120]ps
	M_{delay}	max. delay	[0.6, 1.5]ns
	B_{type}	buffer type	[INV, BUF]
	B_{size}	max. buffer size	x[8, 24]
	B_{tran}	buffer transition	[180, 400]ps
	S_{tran}	sink transition	[150, 380]ps
	M_{levels}	max. levels	[7, 25]
	M_{ww}	max. wire-width	x[1, 2, 3]
	C_{area}	core area	[2, 20]mm^2

For the **CTS** problem, C_{area} and M_{levels} are dependent on M_{sinks}. C_{area} is calculated by assuming that 5% of the total instances are flip-flops, and M_{levels} is calculated as $\log_{FO} M_{sinks}$, where FO is the average fanouts and is assumed to be 10. We generate Design Exchange Format (DEF) [29] files with sinks placed uniformly within C_{area}. We use only DFQD4 cells as sinks. The clock entry point is fixed at the left-bottom corner of the die. We set up the wire width constraints by creating non-default rules (NDR) for all the metal layers in the technology Library Exchange Format (LEF) file [29], and the remaining constraints in the clock tree specification file [30]. We load the design from the DEF, perform clock tree synthesis followed by global and detailed routing by using Cadence *SOC Encounter vEDI10.1* [21]. We extract the clock tree wirelength and total buffer area from these simulations. We use TSMC65PLUS and TSMC45GS cell libraries during clock tree synthesis and routing, and generate a total of 1024 data points.

Figure 1 shows our metamodeling flow. For each type of problem, the golden data points are generated by using the methodology described above. Some of these data points are used to generate samples for the training set; the remainder are used as test data points. We use both LHS and AS to generate the training set. Next, we fit the training data points using the metamodeling techniques. We use academic *MATLAB* [25] toolboxes for MARS [26], RBF [27], and KG [16]. The surrogate models generated by these techniques are used to estimate the responses from the parameters in the test data points. The estimated and the golden responses are compared to calculate the maximum (or worst-case) and average estimation errors for each metamodeling technique and for each response of the three problems.

We describe hybrid surrogate modeling (HSM), a variant of weighted surrogate modeling [7]. In this technique the response is estimated by adding weights to the estimated response from each of the surrogate models. We express this formally as

$$\hat{y}(\vec{x}) = w_1 \cdot \hat{y}(\vec{x})_{MARS} + w_2 \cdot \hat{y}(\vec{x})_{RBF} + w_3 \cdot \hat{y}(\vec{x})_{KG} \tag{10}$$

[3] We could not run *HSPICE* using the TSMC45GS libraries because the device models were not available in the design kit.

where w_1, w_2, and w_3 are the weights of estimated responses of surrogate models for MARS, RBF, and KG respectively. We use least-squares regression to fit the hybrid model to 75% of the training data points which are randomly selected [9]. We perform GCV on the remaining 25% of the training data points to estimate the GCV error. The fitting is repeated 10 times. The weights that give the minimum GCV error out of these 10 tries are used to generate the hybrid surrogate model.

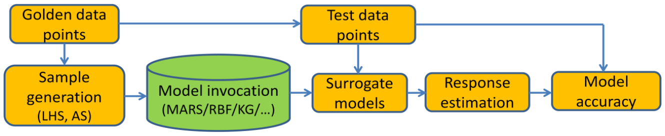

Fig. 1. Metamodeling flow.

V. Results

We conduct experiments according to the methodologies described for each type of problem (NoC, PDN, and CTS) in Section IV. We use the results to study three issues.

- *Impact of sampling strategies and sample sizes.* We compare accuracies of AS over LHS with different N. We also show that increasing N does not always improve accuracy. We empirically show that estimation errors can increase when too many samples are used to derive the model because of overfitting.
- *Impact of dimensionality.* We compare accuracies of metamodels at low- and high-D for the axes outlined in Section I.
- *Metamodeling guidelines for IC design.* Based on our observations, we provide metamodeling guidelines for low- and high-D modeling problems in IC design.

We calculate error by comparing model estimations with golden results from commercial tools. Corresponding correlation coefficients can be estimated as $(100 - error\%)/100$.

A. Impact of sampling strategies and sample sizes

To study the impact of model accuracy on sampling strategies and the number of samples (N), we consider delay estimation of cells under PDN noise at 65nm.[4] We use seven parameters ($D = 7$) in the study to demonstrate the accuracies of MARS, RBF, KG, and HSM techniques. When $D > 7$, KG and RBF have large estimation errors, so, for a fair comparison across all techniques we use seven parameters. We use sampling strategies, LHS and AS, to generate training sets with $N = \{600, 650, 700, 750, 800, 850, 900\}$. We use 5000 data points for testing. Figures 2(a) and (b) show the maximum and average estimation errors of LHS and AS as N varies for each technique. In Tables II and III, we show the percentage reduction in N for the same estimation errors and percentage reduction in estimation errors for the best-case estimates from LHS to AS across all the techniques.

TABLE II
% REDUCTION IN SAMPLE SIZE FOR THE SAME ERROR FROM LHS TO AS

	MARS		RBF		KG		HSM	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
NoC	52.94	52.94	64.71	64.71	52.94	52.94	64.71	64.71
PDN	21.43	21.43	21.43	21.43	11.76	11.76	21.43	21.43
CTS	26.32	26.32	26.32	26.32	24.19	24.19	26.32	26.32

TABLE III
% REDUCTION IN ESTIMATION ERRORS FROM LHS TO AS

	MARS		RBF		KG		HSM	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
NoC	77.42	80.07	72.53	82.63	71.10	66.56	74.82	68.23
PDN	2.63	33.09	34.47	3.48	1.34	10.77	6.83	11.16
CTS	54.70	30.45	52.15	8.25	1.82	5.24	52.53	24.15

Observation 1: AS is *always* better than LHS as it gives smaller estimation errors across metamodeling technique as seen in Table III. The maximum estimation error of AS can be up to 77% less than LHS. This is because uniform sampling by LHS cannot capture nonlinear changes that occur in response to changes in parameters. By using an exploitation-based approach, AS is better able to capture these such nonlinearities in the response in the training set because it uses an exploitation-based approach.

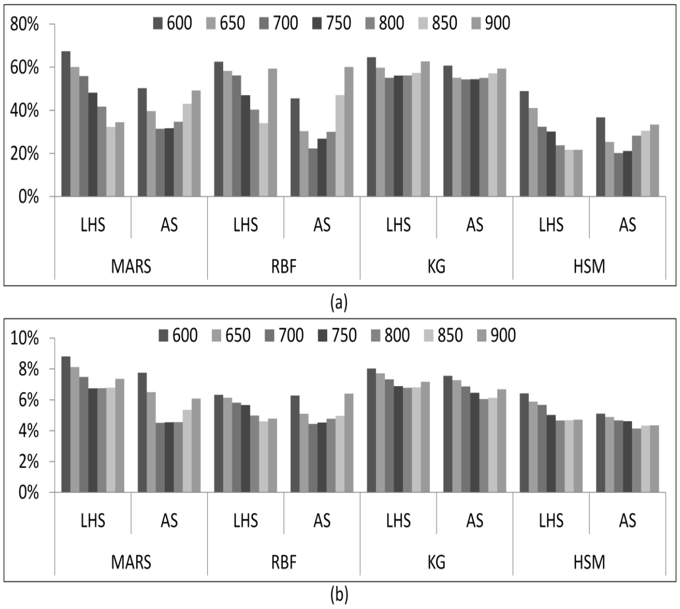

Fig. 2. Comparison of LHS vs. AS across all metamodeling techniques for cell delay estimation under PDN noise with $D = 7$ and $N = \{600, 650, 700, 750, 800, 850, 900\}$. (a) Maximum and (b) average estimation errors. Observe that (1) with $N = 650$, AS gives 33% smaller estimation errors as compared to LHS across all techniques; (2) minimum error occurs with $N = 700$ for AS and $N = 850$ for LHS; and (3) adding samples such that $N > 700$ for AS and $N > 850$ for LHS causes the estimation errors to increase.

Observation 2: AS reduces N by up to 64% to achieve the same estimation errors as LHS as seen in Table II. AS uses exploitation- and exploration-based approaches to add new samples to the training set. It adds samples when the response is sensitive to small changes in the parameters; it does not add samples when the sensitivity of the response is almost constant to changes in the parameters.

Observation 3: Maximum and average estimation errors increase across all techniques for both LHS and AS after reaching a minimum. Adding more than the required number of samples increases the variance in the training set which leads to overfitting in the surrogate models, and causes the GCV errors of the models to increase. As a result, estimation errors increase when test data points are used to predict responses using the surrogate models. The minimum occurs with $N = 700$ for AS, and $N = 850$ for LHS.

B. Impact of dimensionality

To study the impact of dimensionality on model accuracy we demonstrate area and power estimation of NoCs as an instance of low-D modeling problem. We also demonstrate cell delay and output slew estimation under PDN noise, and wirelength and buffer area estimation of clock trees as instances of high-D modeling problems. Here, we report results with only AS since the results in Section V-A empirically demonstrate that AS is always better than LHS.

1. Low-D modeling: NoC

In the NoC power and area estimation problem, we have $D = 5$. We observe that estimation errors are similar at 45nm and 65nm with AS. Hence, we report the average of the maximum and average estimation error values across the two technologies. Figures 3(a) and (b) show the maximum and average estimation errors in area and power.

Observation 4: GPMs are highly accurate for low-D modeling as compared to tree-based models. The maximum estimation error is about 20% for GPMs such as RBF and KG, whereas it is about 30% for tree-based models such as MARS. This is because at low-D, the correlation between parameters is small. We observe VIF values are less than 0.33 for all parameters.[5]

[4] We observe similar trends in estimating output slew under PDN noise, wirelength and buffer area of clock trees, and area and power of NoCs. Owing to space constraints we cannot show all these results.

[5] When VIF values are < 0.33, it indicates the parameters are well-conditioned [2].

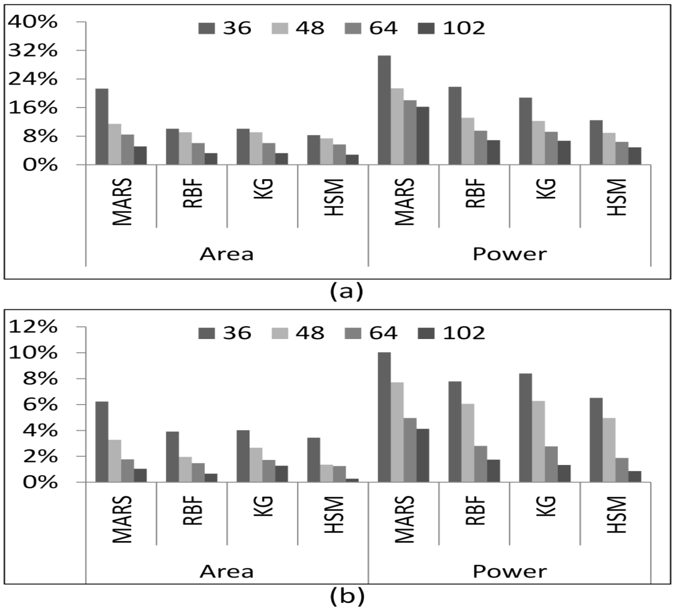

Fig. 3. Comparison of estimation errors for NoC area and power with $N = \{36, 48, 64, 102\}$ generated using AS. (a) Maximum and (b) average.

Observation 5: HSM, in general, outperforms individual surrogate models. For example, HSM can reduce the maximum estimation errors by up to 1.5x as compared to individual surrogate models. For example, HSM reduces the maximum estimation errors in area and power by 8% and 12%, respectively, with $N = 36$.

2. High-D modeling: PDN

In the problem of cell delay and output slew estimation under PDN noise, we have $D = 11$. Each process corner, FF, TT, SS, etc., is characterized by several process parameters such as gate-oxide thickness, channel length, etc. It is difficult to create a unified model that uses the process corner as one parameter because the detailed interactions between process parameters within each corner become obscure. So, we create three models, one for each process corner, and report the average of maximum and average estimation error values. We further use the overdrive voltage, $V_{DD} - V_{TH}$, as one parameter instead of using both V_{DD} and V_{TH}. So, we effectively use $D = 9$ to model this problem. Figures 4(a) and (b) show the maximum and average estimation errors for each technique as D varies. We show these results for $N = 700$ data points generated using AS, and a test set of 5000 data points.

Observation 6: MARS, in general, outperforms RBF and KG in high-D problems. For example, when $D > 7$, multicollinearity [1] significantly increases the variance of the regression coefficients as, and it causes accuracy of RBF and KG to degrade significantly [3], [5], [18]. We observe that VIF values [2] for certain combinations of parameters can be as high as 0.994! More specifically, S, C_ℓ, and $(V_{DD} - V_{TH})$ are highly correlated because increase (resp. decrease) in S can be compensated by increase (resp. decrease) of C_ℓ, or decrease (resp. increase) of $(V_{DD} - V_{TH})$ to give roughly the same cell delay and output slew. RBF and KG are highly sensitive to multicollinearity, whereas MARS is not, because it fits the change in response using piecewise splines. We use ridge regression (RBF+RR in figures) to "cure" multicollinearity, however, as seen from the plots, it is only marginally effective in reducing the large estimation errors of RBF.

Observation 7: Similar to *Observation 5*, HSM can reduce the maximum and average estimation errors by 20% when $D = \{6, 7\}$, and the maximum estimation errors of individual surrogate models are around 35% and have variance larger than 52%. However, when $D = \{8, 9\}$, the average estimation errors of the RBF, RBF+RR, and KG are very high as compared to that of MARS. Therefore, the weight of surrogate model for MARS dominates the weights of other surrogate models. This is the reason the estimation errors of HSM and MARS are the same.

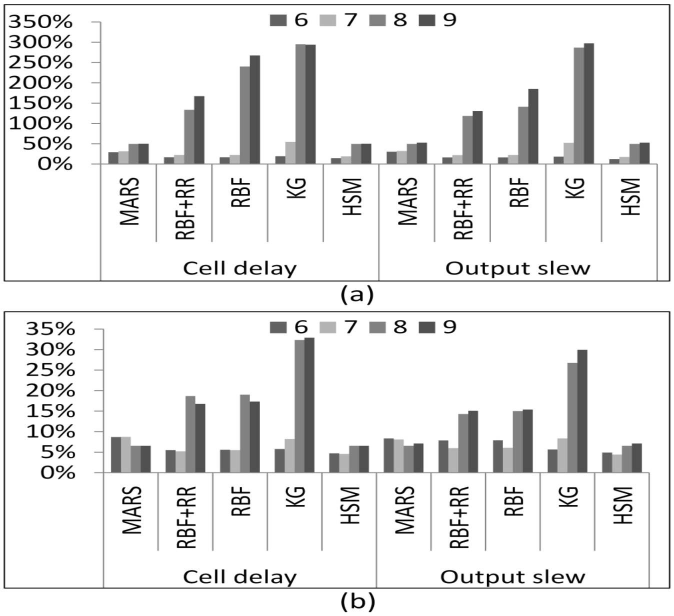

Fig. 4. Comparison of estimation errors for cell delay and output slew under PDN noise at different D with $N = 700$ generated using AS. (a) Maximum and (b) average.

3. High-D modeling: CTS

In the problem of wirelength and buffer area estimation of a clock tree, we have $D = 10$. We observe similar estimation errors at 45nm and 65nm, so we report the average of the maximum and average estimation error values. Figures 5(a) and (b) show the maximum and average estimation errors for wirelength and buffer area. We show these results for $N = 84$ data points generated using AS, and a test set of 1024 data points.

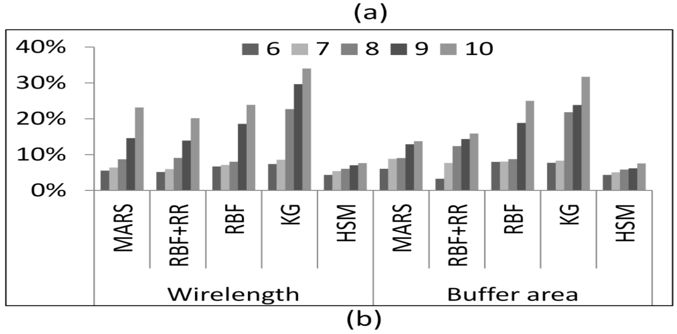

Fig. 5. Comparison of estimation errors in wirelength and buffer area of clock trees at different dimensions with $N = 84$ generated using AS. (a) Maximum and (b) average.

Observation 8: Similar to *Observation 6*, when $D \geq 8$, high collinearity between the CTS parameters causes VIF values to be as high as 0.93, and result in large estimation errors for RBF and KG. More specifically, M_{skew}, M_{levels}, B_{tran}, and B_{size} have VIF values larger than 0.75 when $D = \{9, 10\}$.

Observation 9: Similar to *Observation 5*, HSM reduces the maximum estimation error by up to 3x when the average errors of individual surrogate models are around 30% and have variance of around 46%.

For example, with certain parameters, RBF+RR is more accurate than MARS, and vice versa. Therefore, we obtain weights that minimize the sum of squared errors and the GCV errors $\geq 20\%$ as compared to those of the individual surrogate models.

C. Metamodeling guidelines for IC design

From our observations in Sections V-A and V-B, we provide metamodeling guidelines for IC design that may be useful to architects, design teams, and CAD developers for fast and accurate design space exploration. We propose to always use AS. Figure 6 shows our guidelines in the form of a flowchart. We classify an estimation problem as low-D if $D \leq 5$, else it is high-D. Parameters exhibit low-collinearity if their VIF values are at most 0.33, else they exhibit high-collinearity. For highly collinear parameters, if the average errors of estimates of the surrogate models are at most 30% and the variance in errors is at most 50%, then the problem is classified as *small* μ *and* σ^2, else it is classified as *large* μ *and* σ^2. Based on these classifications our guidelines are as follows

1) *Low-D, low collinearity:* Try RBF and KG. Try HSM with the estimates of these surrogate models.
2) *Low-D, high collinearity:* Try only MARS.
3) *High-D, low collinearity:* Try MARS, RBF, and KG. Try HSM with the estimates of these surrogate models.
4) *High-D, high collinearity, large* μ *and* σ^2*:* Try only MARS.
5) *High-D, high collinearity, small* μ *and* σ^2*:* Try MARS, RBF+RR, RBF, and KG. Try HSM with the estimates of these surrogate models.

We validate these guidelines by using our NoC and CTS problems. The NoC problem with $D = 4$ ends up in the second box in Figure 6 because all the parameters have VIF of 0.27. In the CTS problem, we select $D = 7$ such that all these parameters have VIF values less than 0.33. We obtain wirelength and buffer area estimates from MARS, RBF, and KG models. We then use these estimates in HSM to achieve estimates of wirelength and buffer area that are $\sim 18\%$ more accurate than the estimates from the original surrogate models. This corresponds to the third (from left) box in Figure 6. Our ongoing studies seek further confirmations that our methodology is generalizable to other problems of varying dimensions.

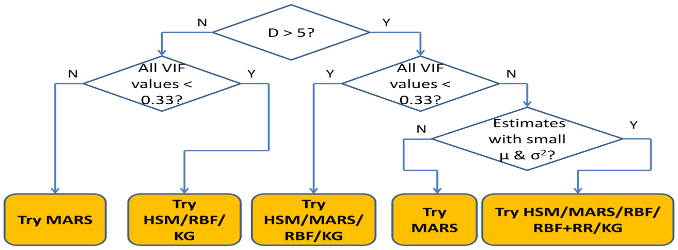

Fig. 6. Metamodeling guidelines for IC design. μ is the mean, and σ^2 is the variance in the estimation errors of each surrogate model.

VI. Conclusions

Metamodeling techniques have recently emerged as effective, low-overhead means for deriving surrogates of physical models in IC design applications. In this work, we use three separate application contexts (NoC, PDN, CTS) to study accuracy limits of the MARS, RBF and KG metamodeling techniques with varying D and N, and with respect to both maximum and average estimation errors. We provide nine observations and insights into the behavior of each modeling technique. Specifically, we observe that for low-D modeling RBF and KG are highly accurate, with worst-case estimation errors less than 30%. However, in high-D contexts these techniques become highly inaccurate, whereas MARS continues to be accurate with worst-case estimation errors within 50%. We also show that newer adaptive sampling strategies are very effective in reducing overheads of sample generation, and in reducing the estimation errors of metamodeling techniques (by at least 20%, when compared against LHS with identical resources). We also document overfitting phenomena (i.e., that having more than the required number of samples in the training set increases variance and adversely affects model accuracy).

We further study HSM, a variant of the method described in [7], which uses weighted combinations of individual surrogate models; our results show that HSM can reduce worst-case estimation errors by up to 3x. Finally, we also provide high-level guidelines for the application of metamodeling techniques to other IC design modeling and estimation problems. Our future work seeks methods to transform ill-conditioned samples to well-conditioned samples, so as to improve accuracies of all metamodeling techniques. We also plan to apply newer machine learning techniques for dimensionality reduction, such as partial least-squares regression, to estimation problems in IC design.

Acknowledgments

Research supported in part by NSF, MARCO/DARPA, Qualcomm and the Semiconductor Research Corporation.

References

[1] D. A. Belsley, “Multicollinearity: Diagnosing Its Presence and Assessing the Potential Damage It Causes Least-Squares Estimation”, *National Bureau of Economic Research Working Paper No. 154*, 1976.
[2] D. A. Belsley, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, Wiley, 1980.
[3] M. D. Buhmann, S. Dinew and E. Larsson, “A Note on Radial Basis Function Interpolation Limits”, *IMA Journal of Numerical Analysis* 30 (2010), pp. 543-554.
[4] C.-K. Cheng, A. B. Kahng, K. Samadi and A. Shayan, “Worst-Case Performance Prediction Under Supply Voltage and Temperature Variation”, *Proc. SLIP*, 2010, pp. 91-96.
[5] N. Cressie, “Geostatistics”, *The American Statistician* 43(4) (1989), pp. 197-202.
[6] K. Crombecq, L. De Tommasi, D. Gorissen and T. Dhaene, “A Novel Sequential Design Strategy for Global Surrogate Modeling”, *Proc. Winter Simulation Conference*, 2009, pp. 731-742.
[7] T. Goel, R. T. Haftka, W. Shyy and N. V. Queipo “Ensemble of Surrogates”, *Struct. Multidiscip. Optim.* 33(3) (2007), pp. 199-216.
[8] D. Gorissen, L. De Tomassi, K. Crombecq and T. Dhaene, “Sequential Modeling of a Low Noise Amplifier with Neural Networks and Active Learning”, *Neural Comput. & Appl.* 18(5) (2009), pp. 485-494.
[9] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
[10] A. A. Ilumoka, “Efficient Prediction of Crosstalk in VLSI Interconnects Using Neural Networks”, *Proc. EPEPS*, 2000, pp. 87-90.
[11] R. Jin, W. Chen and T. W. Simpson, “Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria”, *Struct. Multidiscip. Optim.* 23 (2001), pp. 1-13.
[12] A. B. Kahng, B. Lin and K. Samadi, “Improved On-Chip Router Analytical Power and Area Modeling”, *Proc. ASP-DAC*, 2010, pp. 241-246.
[13] A. B. Kahng, B. Lin and S. Nath, “Explicit Modeling of Control and Data for Improved NoC Router Estimation”, *Proc. DAC*, 2012, pp. 392-397.
[14] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy and S. Borkar, “Near-Threshold Voltage (NTV) Design – Opportunities and Challenges”, *Proc. DAC*, 2012, pp. 1153-1158.
[15] F. Liu, “A General Framework for Spatial Correlation Modeling in VLSI Design”, *Proc. DAC*, 2007, pp. 817-822.
[16] S. N. Lophaven, H. B. Nielsen and J. Sondergaard, “Aspects of the MATLAB Toolbox DACE”, *Technical Report IMM-REP-2002-13*, Technical University of Denmark, 2002.
[17] D. F. Morrison, *Multivariate Statistical Methods*, 3rd edition, McGraw-Hill Publishing Company, 1990.
[18] M. Sarevska, B. Milovanovic and Z. Stankovic, “Reliability of Radial Basis Function – Neural Network Smart Antenna”, *Proc. WSEAS*, 2005, pp. 1-7.
[19] M. B. Yelten, T. Zhu, S. Koziel, P. D. Franzon and M. B. Steer, “Demystifying Surrogate Modeling for Circuits and Systems”, *Circuits and Systems Magazine* 12(1) (2012), pp. 45-63.
[20] T. Zhu and P. D. Franzon, “Application of Surrogate Modeling to Generate Compact and PVT-Sensitive IBIS Models”, *Proc. EPEPS*, 2009, pp. 77-80.
[21] *Cadence SOC Encounter User Guide.* http://www.cadence.com
[22] *International Technology Roadmap for Semiconductors*, 2011, http://www.itrs.net/Links/2011ITRS/2011Chapters/2011Design.pdf
[23] *Synopsys Design Compiler User Guide.*,http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DCUltra/pages/default.aspx
[24] *Synopsys HSPICE User Guide.* http://www.synopsys.com
[25] *MATLAB.* http://www.mathworks.com
[26] *ARESLab.* http://www.cs.rtu.lv/jekabsons/regression.html
[27] *RBF2 Manual.* http://www.anc.ed.ac.uk/∼mjo/rbf.html
[28] *Netmaker.* http://www-dyn.cl.cam.ac.uk/∼rdm34/wiki
[29] *LEF DEF Guide.* http://www.si2.org/openeda.si2.org/projects/lefdef
[30] *Clock Routing Rules.* www.cadence.com/Community/blogs/di/archive 2011/05/10/five-minute-tutorial-setting-up-clock-routing-rules.aspx