Minimizing the Number of Process Corner Simulations during Design Verification

Michael Shoniker, Bruce F. Cockburn, Jie Han and Witold Pedrycz

University of Alberta, Edmonton, AB T6G 2V4, Canada

Abstract — **Integrated circuit designs need to be verified in simulation over a large number of process corners that represent the expected range of transistor properties, supply voltages, and die temperatures. Each process corner can require substantial simulation time. Unfortunately, the required number of corners has been growing rapidly in the latest semiconductor technologies. We consider the problem of minimizing the required number of process corner simulations by iteratively learning a model of the output functions in order to confidently estimate key maximum and/or minimum properties of those functions. Depending on the output function, the required number of corner simulations can be reduced by factors of up to 95%.**

Index Terms — **Adaptive algorithms, circuit simulation, design automation, function approximation, Gaussian processes, robustness, unsupervised learning.**

I. Introduction

To ensure sufficient yield, integrated circuit (IC) designs must be verified in simulation for the expected range of operating conditions. These conditions are specified using process-related parameters (e.g., the threshold voltages and transconductances of the n- and p-type transistors), the supply voltages, and the average die temperature. For convenience the conditions are usually discretized into a finite number of combinations, called "PVT corners." The greater complexity and variability of semiconductor processes at 32 nm and below has increased the required number of PVT corners, with some recent technologies requiring over 1000 corners [1]-[2]. Given that each circuit simulation can be time-consuming, it is desirable to avoid simulating all of the PVT corners (i.e., a "full factorial" experiment) while still ensuring high-confidence design verification (DV).

Verifying circuit behavior involves verifying that key output properties (e.g., voltage, current, delay, rise & fall time) stay within ranges given in the specification. The maximum and/or minimum property values must be determined over all PVT corners. We thus have one or more combinatorial optimization problems for initially unknown function(s) over a discrete input space. Such problems are treated in the subfield of statistics called Response Surface Methodology (RSM) [3]-[4]. If nothing can be assumed about a function, we must verify the function with an exhaustive, full factorial search of the discrete input space. However, if one can make strong (or reasonable) assumptions about the continuity of the function, then it may be possible to guarantee (or to ensure with high probability) design correctness after only a "fractional factorial" search of a subset of the input space. For example, if the output function is known to be "Lipschitz continuous," then the rate of change of the function for any two input vectors will be bounded by some real constant and so search algorithms can safely prune away regions of the search space [5].

II. Conceptual Framework

In our combinatorial optimization problem, the function is expensive to evaluate [6], requiring a long-running simulation. Thus the number of simulations must be minimized. However, given the potentially disastrous consequences of very low yield for an inadequately verified IC, it is costly to terminate the optimization search too soon and then miss the true output maxima (or minima). The overall problem can be decomposed into the three simpler subproblems described in the next three subsections [6].

A. Selecting the Initial Training Set

Selecting the initial training set is the classic experiment design problem in RSM [4],[7]. If there are $n \geq 1$ PVT parameters with distinct maximum, minimum and modal values, then one could include all 3^n joint combinations of these values in the initial training set. Such a 3^n design would allow a fitted quadratic model of the function. However, a 3^n design produces a large training set for $n \geq 5$. A smaller 2^n design can be augmented with inputs to form a "central composite design" that allows a quadratic model to be calculated [7].

We chose to use a heuristic-based modified version of the central composite design (CCD), which picks n^2 initial corners from the 2^n factorial points of the CCD. The training set includes one "typical" corner (with the modal value of each PVT parameter) plus $n^2 - 1$ corners that are selected from the 2^n corners that have extreme values for each PVT parameter. We modified this design using a heuristic that spreads out the $n^2 - 1$ extremal corners. The results of simulating the initial training set corners are used to construct an initial Gaussian Process Model

(GPM) of the output function [8]. We used the software package "scikit-learn" to construct the GPMs [9].

B. Selecting the Next Set of PVT Corners to Simulate

This subproblem chooses the next step (a set of unsimulated PVT corners) in an iterative search of the input space [10]. The objectives could change as the search progresses. Initially the priority might be to learn the global shape of the output function and to identify both the most and least promising input regions. Once candidates for maxima (or minima) have been found, the priority might shift to searching the neighbourhoods of those extrema to help the decision to terminate the search.

Given a set of evaluated input-to-output mappings, a GPM provides for each unevaluated input $\boldsymbol{x}$ an estimate of the corresponding function value $F(\boldsymbol{x})$ plus an estimate of the uncertainty $\sigma_F(\boldsymbol{x})$ of the function value estimate [8],[9]. The uncertainties can be exploited when selecting the next PVT corner(s) to simulate. By combining function estimates with estimates of the uncertainty in those function estimates, one can identify input regions that could contain function maxima, given the maxima found so far. We investigated ways of combining the function estimates and uncertainties to select the next corner(s).

C. Deciding When to Terminate the Search

For each unsimulated PVT corner $\boldsymbol{x}$, the GPM that is computed from the set X of all PVT corners simulated so far provides a function estimate $F_{\text{pred}}(\boldsymbol{x})$ and an estimate of the error $\sigma_{\text{pred}}(\boldsymbol{x})$ in $F_{\text{pred}}(\boldsymbol{x})$ with respect to the actual simulated output $F(\boldsymbol{x})$. The values of $F_{\text{pred}}(\boldsymbol{x})$ and $\sigma_{\text{pred}}(\boldsymbol{x})$ can be used to estimate the likelihood that the simulated value $F(\boldsymbol{x})$ would exceed the largest simulated value $F(\boldsymbol{x}_{\max})$, $\boldsymbol{x}_{\max} \in X$, that has been found so far. We start by assuming that each predicted value $F_{\text{pred}}(\boldsymbol{x})$ will be Gaussian-distributed about the actual output value $F(\boldsymbol{x})$ with standard deviation $\sigma_{\text{pred}}(\boldsymbol{x})$. If this is true, then $\boldsymbol{x}$ could be safely ruled out as the location $\boldsymbol{x}_{\max}$ of the true function maximum if $F(\boldsymbol{x}) + k\ \sigma_F(\boldsymbol{x})$ were to be less than the $F(\boldsymbol{x}_{\max})$ for some suitable k, say $k = 3$, 4 or more.

III. Baseline Simulation Control Algorithm

The baseline algorithm uses the method, described above, for selecting an initial training set of size n^2. In the iterative search, the unsimulated corner $\boldsymbol{x}$ that has the greatest value of $F_{\text{pred}}(\boldsymbol{x}) + 3\sigma_{\text{pred}}(\boldsymbol{x})$ is selected as the next corner to simulate. The search is terminated when no unsimulated corners $\boldsymbol{x}$ remain such that $F_{\text{pred}}(\boldsymbol{x}) + 3\sigma_{\text{pred}}(\boldsymbol{x})$ exceeds the largest output $F(\boldsymbol{x}_{\max})$ found so far. If the GPM does indeed produce normally-distributed predictions of the output function with standard deviation $\sigma_{\text{pred}}(\boldsymbol{x})$, then use of this criterion will terminate the search too soon with a probability of about $Q(3.0) = 0.135\%$, where $Q(d)$ denotes the Q-function which gives the tail probability of a normal distribution at d or more standard deviations from the mean.

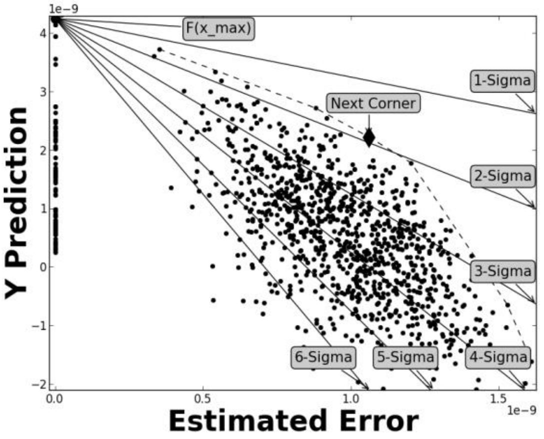

Fig. 1. Convex hull plot for the baseline $F_{\text{pred}}(\boldsymbol{x}) + 3\ \sigma_{\text{pred}}(\boldsymbol{x})$ heuristic.

Fig. 1 illustrates the progress of the algorithm using the baseline $F_{\text{pred}}(\boldsymbol{x}) + 3\sigma_{\text{pred}}(\boldsymbol{x})$ selection heuristic. The function being learned and verified is the fall time of an output of a benchmark circuit provided to us by Solido. This circuit has three outputs that have, to different degrees, proved to be rather difficult to learn. Fig. 1 plots all of the corners $\boldsymbol{x}$ with respect to the (simulated or predicted) output $F(\boldsymbol{x})$ on the vertical axis and the uncertainty $\sigma_{\text{pred}}(\boldsymbol{x})$ on the horizontal axis after 75 (out of 1080) corners have been simulated. The simulated corners appear at the left edge of the plot with uncertainty $\sigma_{\text{pred}}(\boldsymbol{x}) = 0$. Located at the converging point of the lines is the greatest simulated output found so far, $F(\boldsymbol{x}_{\max})$. The remaining unsimulated (and hence predicted) corners appear over a broader region to the right with nonzero uncertainties $\sigma_{\text{pred}}(\boldsymbol{x}) > 0$. The dashed boundary to the top and right of the unsimulated corners is the convex hull of those corners. Intuitively the corners on the convex hull represent "worst-case" corners that could, after simulation, be most likely to exceed the present $F(\boldsymbol{x}_{\max})$ [11]. At each iteration, the one simulated corner $\boldsymbol{x}$ moves to the left edge with uncertainty $\sigma_{\text{pred}}(\boldsymbol{x}) = 0$ and simulated value $F(\boldsymbol{x})$.

The termination heuristic is also illustrated in Fig. 1. The diagonal lines labelled "1-Sigma", "2-Sigma", etc. are defined by the equations $F_j(\boldsymbol{x}_{\max}) - k\ \sigma_{\text{pred}}(\boldsymbol{x}) = 0$, for $k = 1$, 2, etc., where $F_j(\boldsymbol{x}_{\max})$ denotes the largest output value found so far after j simulations. According to the $F_{\text{pred}}(\boldsymbol{x}) + k\ \sigma_{\text{pred}}(\boldsymbol{x})$ heuristic, the termination criterion is met when, for all unsimulated corners $\boldsymbol{x}$, $F_{\text{pred}}(\boldsymbol{x}) + k\ \sigma_{\text{pred}}(\boldsymbol{x}) \leq F_j(\boldsymbol{x}_{\max})$. Thus termination occurs when all unsimulated corners ***x*** lie below the *k*-sigma line. Clearly the algorithm should aim to move the unsimulated corners below the *k*-sigma line in as few simulations as possible. If greater confidence is required that the true $F(\boldsymbol{x}_{\max})$ has been found, then a *k*-sigma line with a larger *k* should be used.

As emphasized above, there is a potentially very large cost to not finding the true maximum (or minimum) of the output(s). One can approximate the probability of finding the true maximum by re-running the simulation control algorithm for many randomly generated choices of the initial training set, and then compute the fraction of those runs that successfully found the true output maximum. Fig. 2 plots the results of such an experiment. Each point represents an averaged value of 100 runs and approximates the probability of finding the true maximum as the total number of simulations increases. Three different output functions–delay, fall time, and rise time–were considered for the same challenging benchmark circuit. (Many other circuits were also considered, but they proved to be easier to learn than the one three-output circuit reported here.) Each run generated a randomized initial training set of n^2 corners from the 2^n factorial points, where n is the number PVT parameters. In this circuit $n = 7$ and the full factorial experiment contained 1080 corners. The points in Fig. 2 have been linked up with lines to show what happens as the simulations are allowed to proceed and meet successively the 3-sigma to 10-sigma termination criteria. The left ends of the lines in Fig. 2 start where the 3-sigma criteria was first satisfied.

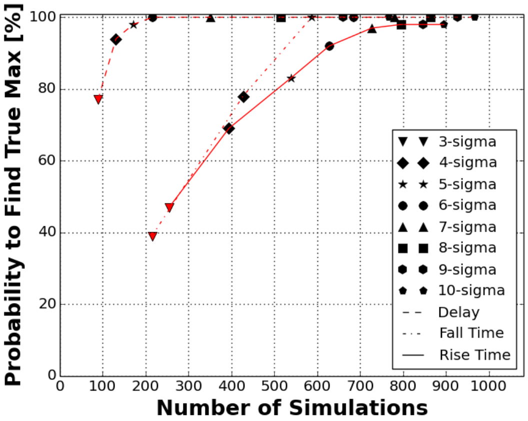

Fig. 2. Probability of finding the true max. using $F_{\text{pred}}(\boldsymbol{x}) + 3\ \sigma_{\text{pred}}(\boldsymbol{x})$.

The plots in Fig. 2 show that the "delay" output (shown with dashed lines) was easier to learn than either the "fall time" (dotted lines) and the "rise time" (solid lines). Note that the 3-sigma termination heuristic was inadequate to ensure that the true maximum was found. For the delay output, it was necessary to meet the 6-sigma termination criterion after 216 simulations to ensure that the true maximum was found for this circuit property. Here, the speed-up using the 6-sigma termination criterion was 1080/216 = 5×.

Fig. 2 shows that the "fall time" and "rise time" outputs were harder to learn than the "delay" output. The fall time output required 587 simulations on average and a 5-sigma termination criterion to always find the true maximum in the 100 runs. The speed-up here was 1080/587 = 1.84×. Once again, the 3-sigma termination criterion failed to find the true maximum with a probability of an ideal 3-sigma of 99.7%. The rise time output was clearly the hardest to learn of the three outputs. For two out of the 100 runs, the true maximum was still not found after 895 simulated corners.

IV. Improved Selection Heuristics

We investigate the performance of an improved search algorithm that applies new heuristics to the baseline algorithm. The first heuristic reduces the initial training size to reduce the number of unnecessary simulations and to allow the algorithm to benefit from more directed selection of corners. The new initial training size is $\max(0.01M, 2n)$, where M is the total number of corners and n is the number of PVT parameters.

From Fig. 2 we see that the rise time output does not rise to 100% probability of finding the true maximum. Upon further investigation, we determined that this termination failure was often due to an inability to detect the true output maximum in an unsimulated corner when that corner was adjacent, in a Manhattan distance sense, from the corner that was already declared by the 3-sigma selection heuristic to be the output maximum. Our study then turned to investigate ways of avoiding the evident weakness in the termination heuristic.

The second heuristic is an enhancement function $E(\boldsymbol{x})$ that magnifies the predicted error $\sigma_{\text{pred}}(\boldsymbol{x})$ to make it more likely that unsimulated corners near the present $F(\boldsymbol{x}_{\max})$ will be considered. $E(\boldsymbol{x})$ has value 1.25 when ***x*** differs from $\boldsymbol{x}_{\max}$ in one discrete step in only one parameter, and value 1.15 when ***x*** differs from $\boldsymbol{x}_{\max}$ in two steps in one or two parameters; otherwise, $E(\boldsymbol{x})$ has value 1.0.

Experiments with a modified search algorithm using the two new heuristics found an increase in the approximate probability of finding the true maximum in the delay and fall time outputs of the circuit of interest, with a 5-sigma termination criterion. However, for the more difficult rise time output, the probability reaches 99% even with the 10-sigma termination criterion.

Based on this result, a third heuristic was further considered to select multiple corners to simulate every iteration. The next corners to be simulated are all the corners that lie on the present convex hull.

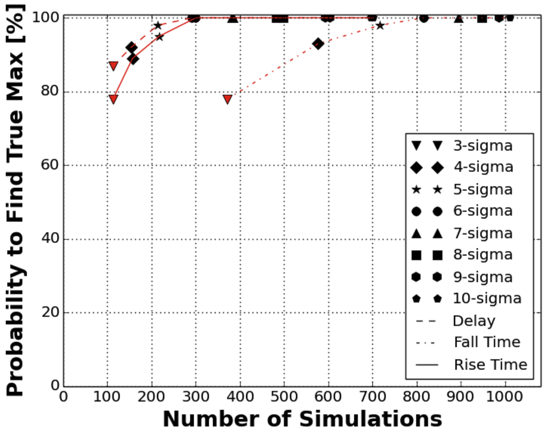

Fig. 3. Probability of finding the maximum using an improved heuristic-based algorithm.

Fig. 3 shows the performance of the improved algorithm. The probability of findiing the true maximum for the fall time and rise time outputs is significantly increased for the 3-sigma stopping criterion. All three outputs reach 100% probability of finding the true maximum by the 6-sigma termination criterion. The difficult rise time output now reaches 100% probability of finding the true max after 300 simulations. The fall time output requires 815 simulations to reach 100% probability. We see a speed up of 1080/291 = 3.71× for the delay output, 1080/300 = 3.6× for the rise time output and 1080/815 = 1.33× for the fall time output.

VI. Conclusions and Future Work

Using Gaussian Process Models to automatically learn approximations of an unknown output function in order to verify a specification for that circuit output proved effective for most but not all of the small industrial benchmark circuits considered. For a set of nine circuits, the convex hull heuristic with a 6-sigma stopping criterion was able to reach speedups from 1.03× to 20.69×; these correspond to reductions in corner simulations ranging from 2.9% to 95%. The average speed up over the nine circuits and forty-six outputs was 4.99×, which corresponds to a reduction of 80% of corner simulations. The convex hull plots proved to be especially useful for visualizing the progress of the learning algorithm towards a termination condition that combined the function values and uncertainties predicted by the constructed GPM for the unsimulated PVT corners.

There are many possible directions for future research. A larger set of typical and harder-to-learn benchmark circuits is being collected to ensure the generality of the new simulation control heuristics. While the GPM approach has proven to be effective in the middle learning phase, it clearly has difficulty learning some output functions and also has difficulty deciding when to terminate. Additional function modeling methods are being used to augment the GPM to overcome its evident weaknesses. Recent work is clarifying how to coordinate the search most efficiently for circuits with multiple outputs.

Acknowledgements

This work was supported by an Engage Grant and by Strategic Project Grant RES0018685 from the Natural Sciences and Engineering Research Council (NSERC) of Canada. We thank Trent McConaghy of Solido Design Automation for his invaluable advice and for benchmark circuit data.

References

[1] K. L. Kuhn et al., “Process technology variation,” *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2197-2208, Aug. 2011.

[2] T. McConaghy, K. Breen, J. Dyck and A. Gupta, Variation-Aware Design of Integrated Circuits: A Hands-on Field Guide, New York: Springer Science+Business Media, 2013.

[3] G. E. P. Box and K. B. Wilson, “On the experimental attainment of optimal conditions,” *J. of the Royal Statistical Society*, Series B, vol. 13, no. 1, pp. 1-45, 1951.

[4] R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, Response Surface Methodology: Process and Product Optimization using Experiments, 3rd ed., Hoboken, NJ: John Wiley & Sons, 2009.

[5] M. Horn, “Optimal algorithms for global optimization in case of unknown Lipschitz constant,” *J. of Complexity*, vol. 22, no. 1, pp. 50-70, Feb. 2006.

[6] D. R. Jones, M. Schonlau, W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *J. of Global Optimization*, vol. 13, pp. 455-592, 1998.

[7] T. P. Ryan, Modern Experimental Design, Hoboken, NJ: John Wiley & Sons, 2007.

[8] C. E. Rasmusssen and C. K. I. Williams, Gaussian Processes for Machine Learning, Cambridge, MA: MIT Press, 2006.

[9] F. Pedregosa et al., Scikit-learn: Machine Learning in Python, *J. of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[10] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Englewood Cliffs, NJ: Prentice Hall, 2009.

[11] A. Gyorgy and L. Kocsis, “Efficient multi-start strategies for local search algorithms,” J. Artif. Int. Res., vol. 41, no. 2, pp. 407-444, May 2011.