Efficient High-Sigma Yield Analysis for High Dimensional Problems

Moning Zhang, Zuochang Ye, Yan Wang
Tsinghua National Laboratory for Information Science and Technology
Institute of Microelectronics, Tsinghua University, Bejing, 100084, China
Email: zhangmn11@mails.tsinghua.edu.cn;{zuochang, wangy46}@tsinghua.edu.cn

***Abstract*—High-sigma analysis is important for estimating the probability of rare events. Traditional high-sigma analysis can only work for small-size (low-dimension) problems limiting to $10 \sim 20$ random variables, mostly due to the difficulty of finding optimal boundary points. In this paper we propose an efficient method to deal with high-dimension problems. The proposed method is based on performing optimization in a series of low dimension parameter spaces. The final solution can be regarded as a greedy version of the global optimization. Experiments show that the proposed method can efficiently work with problems with > 100 independent variables.**

***Index Terms*—High-sigma, yield analysis, importance sampling, high dimension**

I. Introduction

As the feature size of MOSFETs enters sub-28nm realm, process variation caused by random fluctuation of channel dopants, oxide thickness and mobility becomes a limiting factor of transistor scaling. Statistical analysis becomes inevitable to ensure a good production yield.

The most traditional and widely used statistical analysis is Monte Carlo (MC) analysis. The computational cost of MC is inverse proportional to the target failure rate, which, for most circuits, is in the order of 1%. Such method breaks down for applications that require extremely low failure rates, such as $\sim 10^{-6}$. Such problems are usually called "high-sigma" problems, and typical applications include those circuits massively repeated unit cells. As the failure rate of the chip is approximately proportional to the number of cells, and in order for the chip to have a moderate failure rate, the failure rate of the cells must be extremely low.

High-sigma problems in integrated circuits has been studied for several years. Popular methods for tackling high-sigma problems includes mixture importance sampling (MIS) [1], minimum-norm importance sampling (MNIS) [2], Gibbs sampling (GS) [3], Surrogate model assisted importance sampling [4]. These methods work well for SRAM problems, in which the number of transistors, and thus the number of random variables, are small. Typical number of variables in SRAM yield analysis problems are between $6 \sim 24$, depending on how many random variables are considered for each transistors.

In this paper we are tackling a much harder problems by considering > 100 independent variables. Such problems are important for circuits with > 20 transistors, as in the case of D-Flipflops. To the knowledge of the authors, none of existing methods can work with such high-dimensional problems. We have notice that in [5] a 403-dimension case is studied. However, in this work PCA is used to reduce the variables to 11, indicating that 403 is not the number of effective independent variables.

The rest of this paper is organized as follows. Section II gives the mathematical background of the high-sigma analysis problem. Section III introduce the proposed method for tackling high-dimension problems. Section IV gives experiment results. And finally conclusions are drawn in Section V.

This work is supported by National Natural Science Foundation of China (No. 61106031), National 973 Program under Grant 2011CBA00604, 2010CB327403 and Tsinghua University Initiative Scientific Research Program.

978-3-9815370-2-4/DATE14/©2014 EDAA

II. Background

A. Standard Monte Carlo Method

Variational process variables such as threshold voltage V_{th}, oxide thickness T_{ox} and gate length L_{eff} can be characterized as a D-dimensional random variable $X = [x_1, x_2, \ldots, x_D]^T$. Furthermore we can assume its joint PDF $p(X)$ as a multivariate Normal distribution, and each random variables in X can be considered as mutually independent and standard normal [3], i.e.:

$$p(X) = \prod_{i=1}^{D} \left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} x_i^2 \right) \right] \quad (1)$$

The failure rate P_f can be mathematically expressed as

$$P_f = \int_{-\infty}^{\infty} I(X) p(X) dX \quad (2)$$

where $I(X)$ represents the indicator function:

$$I(X) = \begin{cases} 1 & X \in \Omega \\ 0 & X \notin \Omega \end{cases} \quad (3)$$

in which Ω denotes the failure region, i.e. the subset of the variation space where the performance of interest (e.g., read noise margin, write delay for SRAM cells) do not meet the required specification.

Monte Carlo (MC) method is one of the most robust methods for estimating the failure rate P_f. It first generates

N random samples $\{\tilde{X_1}, \tilde{X_2}, \ldots, \tilde{X_N}\}$ according to $p(X)$, then uses SPICE simulation to obtain the corresponding circuit performance $\{\tilde{f_1}, \tilde{f_2}, \ldots, \tilde{f_N}\}$ and hence the indicator function $\{I(\tilde{X_1}), I(\tilde{X_2}), \ldots, I(\tilde{X_N})\}$. Finally, the failure rate is calculated as

$$\tilde{P_f}^{MC} = \frac{1}{N}\sum_{i=1}^{N} I(\tilde{X_i}) \tag{4}$$

For high-sigma applications, MC requires a very large number of samples (e.g. over 10^9) in order to observe a few failure samples. As each sample corresponds to one SPICE simulation, the overall cost is very expensive, or even infeasible.

B. Importance Sampling through Norm Minimization

Importance Sampling has been proposed to overcome the above barriers of standard MC. Its key step is to uses an alternative sampling PDF $q(X)$ to generate M samples. The failure rate can be estimated as

$$\tilde{P_f}^{IS} = \frac{1}{M}\sum_{i=1}^{M} \frac{I(\tilde{X_i})p(\tilde{X_i})}{q(\tilde{X_i})} \tag{5}$$

Compared with standard MC which sampling globally, importance sampling method intends to conduct a more "localized" sampling, aim at including as much failure points with relatively high probability as possible. Literature [2] proposed a norm minimization framework to determine $q(X)$. Its major step is to find the most probable failure point (MPFP) X_{opt} by solving an optimization problem:

$$\begin{aligned} \text{Object :} \quad & \text{minimize } ||X|| \\ \text{Subject to :} \quad & X \in \Omega \end{aligned} \tag{6}$$

Then $q(X)$ is set to be $p(X - X_{\text{opt}})$.

Hence the overall algorithm consists of two stages. The first stage is to solve the above norm minimization problem, and the second stage is sampling with the shifted PDF. To find the global MPFP one needs to search the whole variation space, which always considered as a major problem of this framework [6]. [4] proposed to use surrogate model to substitute SPICE simulation. It first uniformly sampling K points and using SPICE to calculate their corresponding circuit performance, then these points are used for training a surrogate model to replace SPICE simulations.

When the dimension D grows up, the total computation suffers from the so-called "curse of dimensionality", i.e., as variation space's volume grows exponentially with D, search costs for the MPFP will dramatically increases. In addition, as the training samples become sparser and sparser, it is very difficult to construct accurate surrogate models.

III. Robust Method Tackling High-Dimension Application

In this section, we will first introduce a search scheme for the norm minimization problem to reduce the computational complexity from exponential to linear. Then we will show ways to further reduce the required SPICE simulations. Finally we will illustrate that it is always possible to construct a "local" surrogate model for the IS stage and hence give a final hit to the high dimensional application.

A. Subspace Rotation Method for High-Dimension Search

Without loss of generality we consider only the case where the failure region is defined as

$$\Omega = \{X | f(X) < Spec\} \tag{7}$$

where $Spec$ is the pre-given circuit performance specification. And the boundary of the failure region means $\partial\Omega$, as shown in Fig 1.

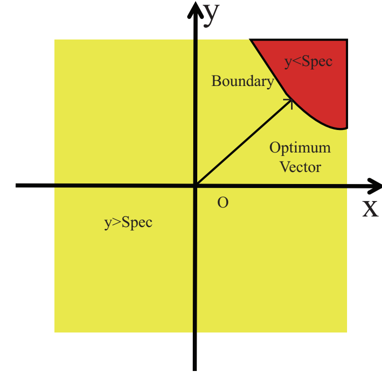

Fig. 1. Geometric notions about variation space.

The key idea of the proposed method is to avoid global search directly in a high-dimension space. Instead, the algorithm performs global search in a series of low dimension subspaces $\{Sub_1, Sub_2, \ldots\}$. To construct such a sequence we first need to define a measurement on each vector starts from the original point, this measure is used to evaluate the closeness to the optimum vector X_{opt}.

Fig 2 shows how the slope of SPICE simulation result reflects variation space's geometric properties. The read current of SRAM, i.e. I_{read}, along 4 different directions are plotted, and the according specification is set as $\text{Spec} = 0.1$, i.e. $I_{\text{read}} < \text{Spec}$ is regarded as a failure. It can be clearly observed that with larger slope, the boundary point is more likely to have smaller norm. Even for cases like (b)(d) where no boundary point is found within the considered range, slope can still distinguish these two cases as we can foresee that (b) will be more likely to have a better boundary point than (d) on a prolonged range.

For practical application,we can simply use the absolute difference between the one-sigma point and the origin point as the measurement:

$$mes(\vec{r}) = |f(\sigma \cdot \frac{\vec{r}}{||r||}) - f(O)| \tag{8}$$

With these definitions we can now write the pseudo code of our proposed search scheme as in Algorithm.1

The proposed algorithm works by first sorting all the D dimensions according to their measurements in a descending

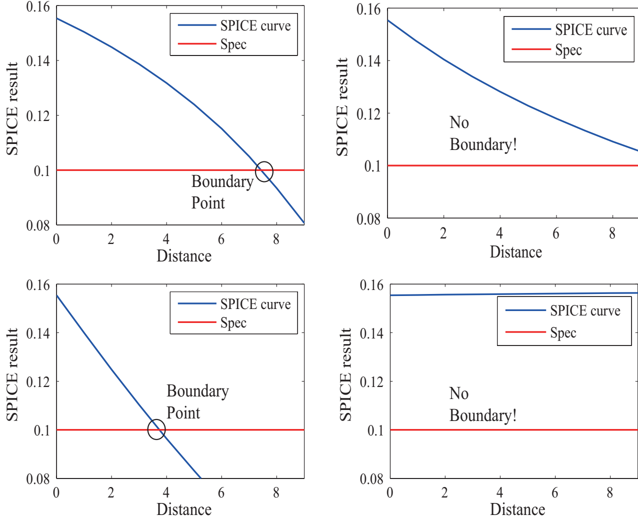

Fig. 2. How SPICE curve's slope reflect geometry structure of variantion space.

Algorithm 1 Framework of Subspace Rotation Method

$Dim = D, Subdim = d$
$r = 1;$
$V :=$ Identitymatrix(D);
$mes(V) = \{mes(V(1)), \ldots, mes(V(D))\}$;
$V =$ sort$(V,$ 'descending according to $mes(V)'$);
while $r \leq D$ **do**
  if $r + d - 1 < D$ **then**
    $Base := \{V(r), V(r+1), \ldots, V(r+d-1)\}$;
    $Sub :=$ Subspace$(Base)$;
    $V(r+d-1) =$ OptVector(Sub);
    $r = r + d - 1$;
  else
    $Base := \{V(r), V(r+1), \ldots, V(end)\}$;
    $Sub :=$ Subspace$(Base)$;
    $V(end) =$ OptVector(Sub);
    break
  end if
end while

manner which in fact is an order about how effectively each variable affects circuit performance. During each iteraten we successively introduce d variables into consideraten and after the optimization we replace these d vectors with the optimum vector found in the subspace formed by the d variables. Thus the whole procedure looks like rotating a d-dimensional subspace while trying to capture the global MPFP in it.

It is easy to verify that this framework is greedy with computation complexity $O(d)$, and it can be considered well escaped from the "curse of dimensionality". In addition, as the algorithm sorting in a descending order according to the measurement, we can usually accelerate the converge speed by set proper criteria to stop the iteraten when the rest dimensions make only ignorable effect on circuit performance.

B. Fast Low-Dimension Method Through Interpolation

In the above discussion have reduce the high dimension search issue into a series of low dimension problems. The rest of the task is to deal with the global optimization in low-dimension spaces. To effectively solve the low dimension case we can borrow tools from interpolation theory.

If we choose $d = 2$, the variation space can be viewed as a square bounded by $X^- < X < X^+$ (e.g. $-9\sigma \sim +9\sigma$). Firstly we discretize the variation space into uniform meshes as a $N \times N$ matrix, and then randomly choose K mesh points, after SPICE simulation we can obtain a sequence of points $P_{known} = \{X_1, X_2, \ldots, X_k\}$ with known circuit performance. The aim is to recover a continuous function's response surface from these few known points, which is just similar to the task of recovering a smooth image from few known pixels. One of the simplest yet robust method for image recovery is to minimize the so-called $\ell 2$-norm of total variation [7]:

$$\begin{aligned} &\text{Object: minimize } TV = \\ &\sum_{t_1,t_2}[(g(t_1,t_2)-g(t_1-1,t_2))^2+(g(t_1,t_2)-g(t_1,t_2-1))^2] \\ &\text{Subject to: } g(P_{known}) = f(P_{known}) \end{aligned} \tag{9}$$

here $g(X)$ is the requiring interpolation result of SPICE function $f(X)$.

The most important advantage of the TV method is that the optimization problem is convex, which can be quickly solved with robust convex optimization methods like Interior-point method [8]. To show the accuracy of this interpolation, we conduct experiment on four different circuit performances: read current I_{read}, write noise margin WNM, static noise margin SNM and read noise margin RNM of the SRAM cell. The considered range of variation space is from $-9\sigma \sim +9\sigma$ and discretized as 91×91 uniform meshes, the number of known points is set as $K = 30$. Finally the resulted mean-square error and corresponding specification is shown in Table I. The above result shows that this interpolation method can provide $\sim 10^{-4}$ accuracy, which can be considered accurate enough for practical application.

TABLE I
INTERPOLATION ACCURACY.

Performance	Mean-square error	Spec
I_{read}	2.013×10^{-4}	0.1
WNM	1.02×10^{-4}	0.05
SNM	1.15×10^{-4}	0.02
RNM	1.02×10^{-3}	0

Theoretically speaking, the above method can be extended to a larger d. Considering implementation issue, one must strike a balance between storage and speed. Fortunately, $d = 2$ with above experiment parameters already offers us a reasonable costs. For application with more than 100 dimensions, roughly 3000 runs of SPICE is enough, and for

low dimensional problem like $D = 6$, less than 300 runs are required.

C. IS Using Surrogate Model

Compared with the norm minimization stage, importance sampling is a more "localized" process which means we do not need to have the knowledge of a whole D-dimension space. Furthermore, [4] proposed that by using fine surrogate model, IS can be accurately done even without any SPICE simulations.

For most fitting methods, a surrogate model is represented as linear combination of some basis functions $\mathbf{B}(X)$ [9]:

$$f(X) = a_0 + \sum_i a_i * B_i(X) \tag{10}$$

By Taylor expansion, the SPICE function $f(X)$ can be locally approximated around X_{opt} as

$$f(X) = f(X_{\text{opt}}) + \mathbf{c_1}\Delta\mathbf{X} + \mathbf{c_2}\Delta\mathbf{X}^{\otimes 2} + \ldots \tag{11}$$

Thus the "localized" property of IS stage enables us to use simple basis functions like linear or quadratic terms.

Some useful tools such as Lasso [10] can help to improve the quality of the surrogate model because of its ability of variable selection for high-dimensional case. Instead of just focusing on fitting accuracy, Lasso add coefficient terms into minimization object function, helping to generate a less sensitive model [9]:

$$\mathbf{a}^* = \text{minimize } ||\mathbf{y} - \mathbf{B}(X) * \mathbf{a}||^2 + \lambda_2||\mathbf{a}||^2 + \lambda_1||\mathbf{a}||_1 \tag{12}$$

With the help of local surrogate model, we can always reach desired accuracy with no more than 10^2 level of SPICE runs, much less sensitive to the increase of dimension D.

D. Summarize

Finally, we can summarize the proposed failure rate calculation method in Algorithm 2.

Algorithm 2 Realization of Failure Rate Estimation

Step 1: Sort all D variables as an descending queue according to their measurements;
Step 2: Select the first d vectors in the queue as bases, if the total length is less than d, select all of them.
Step 3: Discretized the d-dimension subspace formed by the picked vectors, then randomly choose K points in it and run SPICE simulation;
Step 4: Solving convex optimization problem (9) and obtain the optimum vector acoding to norm minmization criteria;
Step 5: Substitute the first d vectors with the optimum vector obtained above and goto Step 2.
Step 6: Draw T points randomly according to PDF $q(X) = p(X - X_{\text{opt}})$ and run SPICE simulation;
Step 7: Using these T points to train a surrogate model;
Step 8: Generate M points from $q(X)$, use surrogate model to evaluate them;
Step 9: Calculate failure rate P_f according to eq.(5);

IV. Results

In this section, the 6-T SRAM cell designed in a commercial 40 nm process is used to demonstrate the accuracy and efficiency of the proposed method. Fig 3 shows the schematic of the SRAM cell. In total, 17 variation variables such as T_{ox} and V_{th} mismatch of each transistor are considered (thus $D = 6 \times 17 = 102$). All the process variables are treated as mutually independent and standard normal. The circuit performance is chosen as the read noise margin (RNM) to evaluate the stability of the SRAM cell [11]. When RNM is less than zero, the data retention failure happens, so we set $Spec = 0$. For comparison, all of the following experiments conducted respectively on four SRAM cells with different design parameters labeled as SRAM1 $\sim$ SRAM4.

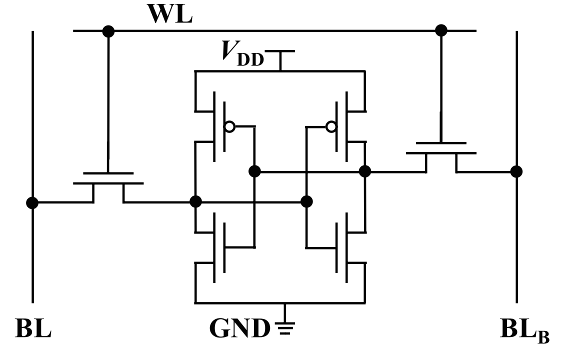

Fig. 3. The circuit schematic of the 6-T SRAM cells.

A. Procedure of Subspace Rotation Method

As been discussed, the proposed subspace rotation method intends to find a variation plane (when $d = 2$, otherwise should be called subspace) which contains the global MPFP. To do this, the plane is rotated iteratively with a greedy scheme, in this example, totally $D - 1 = 101$ iterations are played. Thus we can plot the boundaries of these 101 planes and fold them together into one figure to verify the efficiency of this scheme. In Fig 4 the first 4 runs are being plotted and in Fig 5 the last 4 are observed.

It can be clearly observed that the boundary is moving towards the original point with growing iterations, which means we are getting better MPFP and hence verify the effectiveness of the proposed rotation scheme for our norm minimization task. Also we see that boundary moves rapidly during early iterations but almost come to a halt at later runs, so actually we don't need full $D - 1$ iterations to acquire convergency, when such kind of stop is observed we can safely break the iteration because we initially use a descending order based on their measurements.

B. Validate the Accuracy of the Proposed Method

To evaluate the accuracy of the proposed method, standard Monte Carlo method on the above four SRAM cells are being compared. The failure ratio P_f of these SRAM cells vary from $10^{-5} \sim 10^{-3}$ level, thus the costs of SPICE simulation for

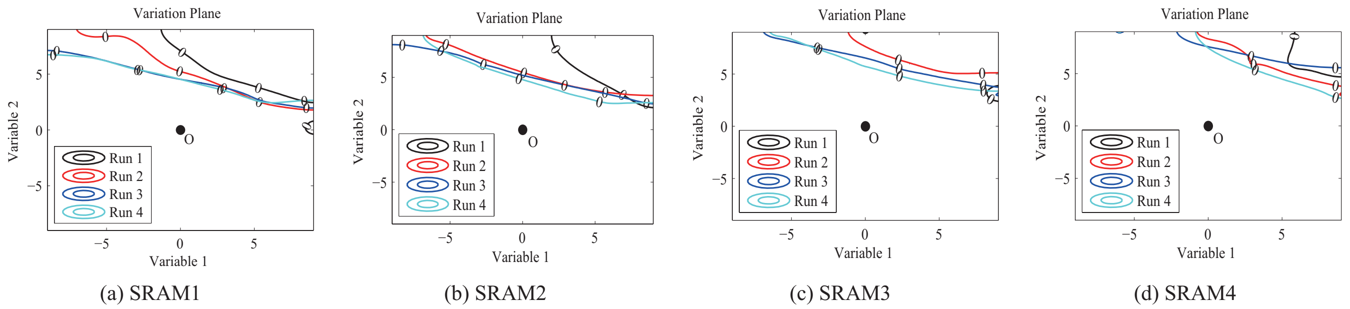

Fig. 4. How boundary moving towards the original point during the first four iterations for the four SRAM examples. (Variable 1 and variable 2 represents the bases of subspace in different iterations, they are transformed into the same plane for vision simplicity)

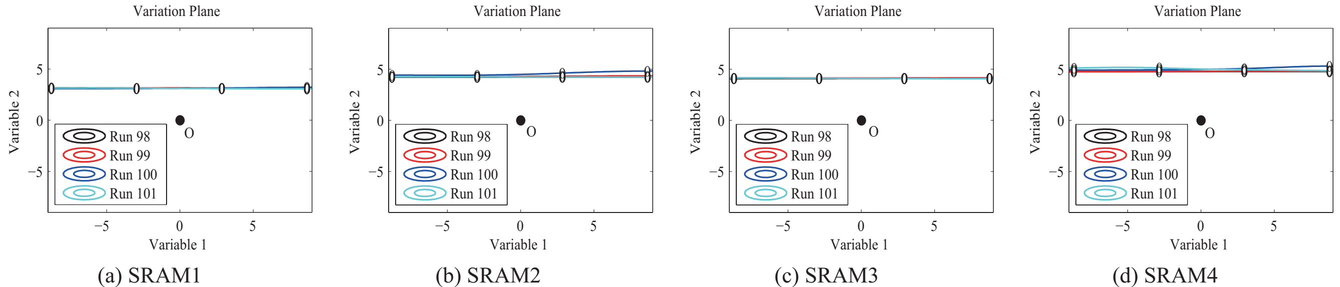

Fig. 5. How boundary moving towards the original point during the last four iterations for the four SRAM examples.(Variable 1 and variable 2 represents the bases of subspace in different iterations, they are transformed into the same plane for vision simplicity)

standard Monte Carlo should be no less than $10^4 \sim 10^6$ to ensure convergency.

In contrast, the proposed method only 3235 runs of SPICE simulations are required (i.e. $d = 2$, and 205 simulations for sorting measures, $101 \times 30 = 3030$ simulations for subspace rotation method in the worst case) for the norm minimization stage. While for the IS stage, 200 runs of SPICE simulations are used for training a local surrogate model, and the rest sampling and circuit performance calculation job are done with this model within less than 1s. Thus with the proposed method, no more than 3500 SPICE runs are needed to obtain the same accuracy as standard MC, reaching about 100X∼1000X acceleration. Fig 6 plots failure rate against simulation runs for the proposed method and standard Monte Carlo. The failure rate estimation from the two methods closely match each other, validating the accuracy of the proposed method.

C. Compare with Existing Methods

The main goal of the proposed method is to tackle high-dimensional applications. The most severe problem of the existing methods using norm minimization and IS framework is the "curse of dimensionality" when searching MPFP using global optimization. One of the fastest method under this framework found in references was proposed in [4], which used a global surrogate model using radial basis function network [12], combined with differential evolution algorithm (DE) [13] to find MPFP. For reasons we've discussed in Section II, both the construction of a global surrogate model and the direct-search scheme greatly suffer from large D. Table II compares the norm of the resulted MPFP using this method and our proposed one on the four different SRAM cases.

TABLE II
NORM OF RESULTED MPFP($\times\sigma$).

Label	Surrogate Model+DE	Proposed
SRAM1	8.5137	3.225
SRAM2	7.8606	3.8418
SRAM3	7.7157	4.2426
SRAM4	9.6831	5.5569

It can be clearly observed that the Surrogate Model+DE method will result in a bad MPFP (deviated more than 3σ) for high dimensional application, leading to a poor distorted sampling PDF $q(X)$, and consequently make large estimation error.

Other methods tackling high-dimension application without using this framework have also been proposed, but they've only reported to solve cases up to about 20 random variables. For example, [14] deals with 12-d analysis, [5] considers a 403-dimensional case but using primary component analysis to reduce the dimension to 11, and [6] solves a 24-dimensional case. Our proposed method makes no assumption on the intrinsic dimension of the process variables and the topology structure of the circuit schematic, thus can be well applied to

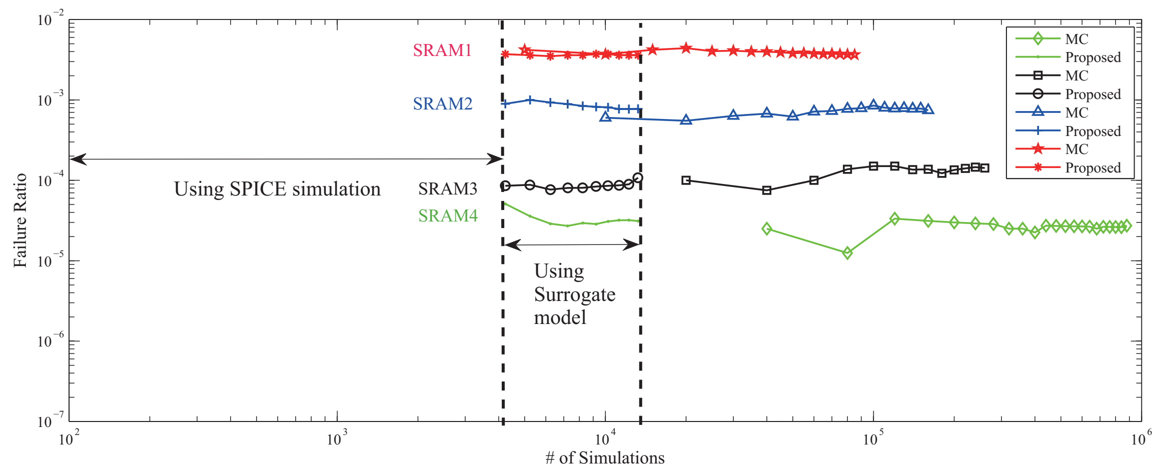

Fig. 6. The estimated failure rate as a function of the total number of samples by the proposed method and standard Monte Carlo method for the four SRAM examples.

circuits other than SRAM.

For low dimensional cases, the SPICE simulation costs of the proposed method are about the same level of the best known methods. For example, with $D = 6$ application, no more than 300 runs of SPICE are required, even slightly less than the best result reported in paper [4], according to which is far better than the existing MIS [1], MNIS [2],GS [3] methods which all require more than 2000 runs of SPICE simulations. Thus the proposed method can well compete with the best known method for low-dimension case while also remain feasible for very high dimensional applications, which can not be well solved with other existing methods.

V. Conclusions

In this paper we proposed an efficient and robust method for high-dimensional high-sigma analysis problems. Existing methods such as the norm minimization importance sampling method suffer from cost explosion when finding the optimal failure boundary due to "curse of dimensionality". We proposed a space rotation method to convert the high-dimension optimization problem to a series of low-dimension problems. Each low-dimension problem is formulated as a convex optimization problem and thus global optimality can be obtained. The proposed method can easily handle problems with > 100 random variables.

References

[1] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *ACM/IEEE Design Automation Conference*, Jun. 2006, pp. 69 –72.

[2] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: Sram evaluation through norm minimization," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2008, pp. 322 –329.

[3] C. Dong and X. Li, "Efficient sram failure rate prediction via gibbs sampling," in *ACM/IEEE Design Automation Conference*, Jun. 2011, pp. 200 –205.

[4] J. Yao, Z. Ye, and Y. Wang, "Efficient importance sampling for high-sigma yield analysis with adaptive online surrogate modeling," in *Design, Automation Test in Europe Conference Exhibition*, March. 2013, pp. 1291 – 1296.

[5] A. Singhee and R. Rutenbar, "Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application," in *Design, Automation Test in Europe Conference Exhibition*, Apr. 2007, pp. 1 –6.

[6] K. Katayama, S. Hagiwara, H. Tsutsui, and H. Ochi, "Sequential importance sampling for low-probability and high-dimensional sram yield analysis," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2010, pp. 703 – 708.

[7] T. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematic, 2005.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[9] T. McConaghy, "High-dimensional statistical modeling and analysis of custom integrated circuits," in *IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2011, pp. 1 – 8.

[10] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Press, 2011.

[11] K. Agarwal and S. Nassif, "The impact of random device variation on sram cell stability in sub-90-nm cmos technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 1, pp. 86 –97, Jan. 2008.

[12] L. M. Surhone, M. T. Timpledon, and S. F. Marseken, *Radial Basis Function Network*. VDM Verlag Dr. Mueller e.K., 2010.

[13] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[14] M. Qazi, M. Tikekar, L. Dolecek, and D. Shah, "Loop flattening & spherical sampling: Highly efficient model reduction techniques for sram yield analysis," in *Design, Automation Test in Europe Conference Exhibition*, March. 2010, pp. 801 – 806.