Bayesian Optimized Importance Sampling for High Sigma Failure Rate Estimation

Dennis D. Weller, Michael Hefenbrock, Mohammad S. Golanbari, Michael Beigl and Mehdi B. Tahoori
Karlsruhe Institute of Technology, Karlsruhe, Germany
{dennis.weller,michael.hefenbrock,mohammad.golanbari,michael.beigl,mehdi.tahoori}@kit.edu

***Abstract*—Due to aggressive technology downscaling, process and runtime variations have a strong impact on the correct functionality in the field as well as manufacturing yield. The assessment of the yield and failure rate is extremely crucial for design optimization. The common practice is to use Monte Carlo simulations in order to account for device variations and estimate failure rate. However, Monte Carlo methods are infeasible for estimating rare events such as high sigma failure rates, and hence, various importance sampling methods have been proposed. In this paper, we present an efficient importance sampling approach based on Bayesian optimization. Its advantages include constant complexity independent of the dimensions of design space, the potential to find the global extrema, and higher trustworthiness of the estimated failure rate. We evaluated the approach on a 6T SRAM cell based on a 28nm FDSOI process. The results show significant speedup and more than two orders of magnitude better accuracy in failure rate estimation, compared to the best state-of-the-art technique.**

I. Introduction

As the transistor density in integrated circuits rapidly increases, it is imperative to decrease the failure probability per component to ensure high manufacturing yield and reliable operation in the field. At the same time, due to nanoscale effects, the device level reliability is significantly reduced due to higher process and runtime variations [1]. One crucial component in this regard are memory blocks, which occupy a large portion of the modern Systems on Chips (SoCs) and continue to grow in size as per International Technology Roadmap for Semiconductors (ITRS) [2]. The failure rate of the individual memory components such as SRAM cells has to be limited within a very small margin to enable proper memory operations. The conventional Monte-Carlo (MC) approach to perform a statistical failure analysis in a circuit design environment is to execute many simulations based on a statistical model of the circuit and evaluate the failure rate. A reasonable failure rate estimation can only be achieved with a very high number of simulations, especially when the expected failure rate is small. This makes the conventional MC approach so computationally expensive that is almost infeasible.

Several techniques have been proposed in the literature with the aim of permitting the MC method for rare event estimation. Some of the existing methods such as purely analytical methods [3], [4] or hybrid methods [5], [6] may not be suitable for modern complex components such as memory blocks because they cannot incorporate various variation parameters or they need a specific model of the problem. In this regards, Importance Sampling (IS) methods [7]–[12] are shown to be very potent and popular to address this problem.

In IS methods, MC samples are generated from the border of the failure region by shifting the MC sampling distribution. This leads to a significant increase in the number of observed failures, which reduces the number of required MC samples by some orders of magnitude. Then, the failure rate calculated based on the simulation results is adjusted based on the shift of the distribution. IS methods can potentially estimate an accurate failure rate with high accuracy by a limited number of simulations. However, there are at least two concerns regarding the IS methods. The first concern is still about the feasibility of the IS methods, especially for high sigma failure rate estimation. Therefore, new methods offering higher accuracy at lower number of samples are required due to the growing demand for larger memory size and lower failure rates per cell. The second concern is about the trustworthiness of an IS method. When the sampling distribution is shifted to a proposed failure region in the parameter space, other failure regions may be overlooked, which can lead to an underestimation of the failure rate [13]. Therefore, finding a correct shift in the MC sampling function, which is aware of multiple critical failure regions is crucial.

In this paper, we propose a *Bayesian Optimization based Importance Sampling approach* (BOIS), which aims at addressing the aforementioned concerns. Bayesian Optimization (BO) is an efficient approach for global optimization of a black-box function with minimum number of evaluations. As BO tries to find the optimum of the objective function, a probabilistic surrogate model of the function is created and refined for each new evaluation.

Then, the shift in the MC sampling function is extracted based on the developed surrogate model. The main advantage of this approach is that once the surrogate model is elaborated enough, the extracted shift in the MC sampling function is located in a region with high probability of failure. As a result, the proposed IS method converges very fast with high accuracy.

The results of our simulation analysis on a commercial 6T-SRAM cell design demonstrate a speedup of at least $2\times$ compared to the state of the art [12] for small sigma problems (rare-events). In high sigma problems (extremely rare events), our proposed method does not overlook significant failure regions, therefore, the estimated failure rate of our method is correctly $632\times$ larger than the state of the art with the same number of simulations [12], which overlooks other important failure regions. Therefore, our proposed method exhibits superior speedup and accuracy compared to the state of the art.

The rest of this paper is structured as follows: In Section II, we describe the context of this work and report on the state-of-the-art techniques, developed to date. Our approach is presented in Section III, where the proposed BOIS framework

978-3-9819263-2-3/DATE19/©2019 EDAA

is introduced. Our evaluations are covered in Section IV and Section V concludes this work.

II. Preliminaries

A. Failure Rate estimation using Monte Carlo methods

For the analysis of the failure probability of a circuit, the impacts of variations in the design parameters of the components have to be considered. Such design parameters could, for example, be threshold voltages of transistors, capacitances, or resistances [11]. Depending on the combinations of those parameters, different circuit characteristics can be observed. Different types of failures, such as timing or functional failures, may happen due to the variation impacts. For example, the circuit delay may exceed a certain threshold leading to timing failures.

In the following, we denote the random vector $\vec{x}$ as the vector of random variables x_i, where x_i denote standardized differences to the nominal parameter value of i-th variation parameter of a circuit. Without the loss of generality, the individual variational parameters x_i may be assumed as independent and identically distributed[1] (i.i.d.) with $p(x_i) = \mathcal{N}(x_i; 0, 1)$, i.e. standard normally distributed. Furthermore, as we want to quantify the failure rate, e.g. due to timing violations, we denote f_{crit} as the critical value of the delay $f(\vec{x})$. If a configuration $\vec{x}$ with $f(\vec{x}) > f_{crit}$ is observed, we classify this as a circuit failure. The *failure rate* (P_f) is expressed as:

$$P_f = \int_X \mathbb{1}(f(\vec{x}) > f_{crit})\, p(\vec{x})\, d\vec{x}, \tag{1}$$

where X denotes the parameter space, and

$$\mathbb{1}(f(\vec{x}) > f_{crit}) = \begin{cases} 1, & \text{if } f(\vec{x}) > f_{crit} \\ 0, & \text{if } f(\vec{x}) \leq f_{crit} \end{cases}$$

denotes the indicator function. As no closed-form expression of P_f is available, and P_f can be written as an expected value of the indicator function, P_f can be estimated using Monte Carlo (MC) Methods. The MC estimator $\hat{P}_f$ of P_f can be:

$$\hat{P}_f = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(f(\vec{x}_i) > f_{crit}),$$

based on N samples drawn from a sampling distribution $p(\vec{x})$.

The convergence rate ρ is an important Figure of Merit (FoM) [14] for MC techniques, which represents how fast an MC technique reaches to a specific accuracy. Smaller convergence rates are preferred, as they lead to a higher accuracy for a given number of N simulations. The FoM can directly be computed from the variance and expectation of the failure rate estimate $\hat{P}_f$, given by (2).

$$\rho(\hat{P}_f) = \frac{\sqrt{Var(\hat{P}_f)}}{\hat{P}_f} \tag{2}$$

As the probability P_f is expected to be small, i.e. failures represent rare events, usually a lot of samples are required to achieve a reasonable accuracy [14]. Since evaluating $f(x)$ for each sample x is costly, a conventional MC technique is computationally infeasible.

[1] Correlated parameters may be mapped into i.i.d. using techniques such as Principal Component Analysis (PCA).

B. Importance Sampling

The aim of the Importance Sampling (IS) techniques is to reduce the number of required samples for a reasonable accuracy and make the MC method feasible. To increase the number of observed failures, and thus the convergence rate, one can sample from a different (sampling) distribution $q(\vec{x}; \vec{\mu}_q)$. This other sampling distribution may represent a shifted variant of $p(\vec{x})$, i.e. a standard normal distribution with mean $\vec{\mu}_q$, that samples closer to the high delay region and results in $f(\vec{x})$ values closer to f_{crit}.

While IS increases the number of observed failures, it will also lead to a biased sampling of $p(\vec{x})$. To compensate for this, it is necessary to adjust for the probability of observing $\vec{x}$ under $q(\vec{x}, \vec{\mu}_q)$ by a correction term $\frac{p(\vec{x})}{q(\vec{x}, \vec{\mu}_q)}$, which is commonly referred to as weight-factor $w(x)$, as seen in (3). Therefore, the failure rate evaluated by IS ($\hat{P}_{f,IS}$) is:

$$P_f = \int_X \mathbb{1}(f(\vec{x}) > f_0)\, \frac{p(\vec{x})}{q(\vec{x}; \vec{\mu}_q)}\, q(\vec{x}; \vec{\mu}_q)\, d\vec{x} \tag{3}$$

$$= \int_X \mathbb{1}(f(\vec{x}) > f_0)\, w(\vec{x})\, q(\vec{x}; \vec{\mu}_q)\, d\vec{x} \tag{4}$$

$$\hat{P}_{f,IS} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(f(\vec{x}_i) > f_{crit})\, w(\vec{x}) \tag{5}$$

Although the application of IS is straight-forward, the choice of a suitable sampling distribution $q(\vec{x}; \vec{\mu}_q)$, i.e. choosing $\vec{\mu}_q$ is non-trivial. In fact, every IS method follows a specific path to find a good $\vec{\mu}_q$.

C. Related Work

Many research studies have been conducted on how to run IS accurately for estimating the failure probability. For instance, [7] reports on an efficient way for estimating the failure rate, however, still the total number of simulations is high. Recently, [12] proposed a very efficient and accurate approach of estimating the failure rate. Their approach is divided into two steps: first, gradient descent enforced simulations are performed for finding the most probable failure point (MPFP), also called optimal shift vectors (OSVs). The MPFP is a design point for which the circuit fails, but is still very close to the origin of the problem space, thus increasing the likeliness of failures at this point. Subsequently, in a second step, importance sampling is applied, where the new sampling function is computed by the original sampling function by shifting the mean by the MPFP, thus enabling sampling from the failure region. While this approach outperforms existing methods, it has the drawback of having linear complexity with respect to the dimensions of $\vec{x}$. Moreover, due to the gradient descent routine, it cannot solve for non-convex problems and the obtained MPFP might not be the global optimum. In [8] an algorithm was introduced, which can reduce the number of dimensions, nevertheless, it requires a pre-sampling phase with many thousands of Monte Carlo simulations. The authors in [9] went one step further, as they present a technique with constant time complexity. Nevertheless, prior knowledge

about the circuit is required, which makes it inappropriate for generalization. [10] presented a method independent from the number of dimensions, which reduces the simulation time. However, a user-defined parameter has to be added to guarantee proper operation. In particular, this parameter scales the variance of $p(\vec{x})$ in order to cover the failure points by using Scaled-Sigma Sampling (SSS). However, this is not applicable for high-dimensional problems as sampling is distributed to the entire problem space, and sampling from the failure region becomes less likely. In [11], the authors report on a Bayesian optimized approach, where the unknown function $f(\vec{x})$ is approximated by a Gaussian process. But the solution is again based on SSS, which is not very efficient.

In summary, the existing IS methods suffer from local optimization and inefficiency, particularly for high-dimensional problems. We will address these shortcomings in the proposed BOIS approach.

III. Bayesian Optimization based Importance Sampling

This section presents the proposed BOIS methodology. Without the loss of generality, we use an exemplary SRAM cell timing failure, in which an increase in the delay of an SRAM cell beyond a certain threshold f_{crit} leads to a timing failure. Therefore, the aim of the BOIS method is to determine the timing failure rate of each SRAM cell with high accuracy and minimum number of simulations.

Since we exploit the IS methodology, we need to find a suitable sampling distribution $q(\vec{x}, \vec{\mu}_q)$, i.e. shift vector $\vec{\mu}_q$, to improve the convergence rate and accuracy. A good choice of $\vec{\mu}_q$ could be the border of a failure region, because the sampling distribution $q(\vec{x}, \vec{\mu}_q)$ would contain a fair number of failure samples. We define the *delay-margin* function as:

$$\text{Delay-margin:} \quad g(\vec{x}) = (f(\vec{x}) - f_{crit})^2, \tag{6}$$

so that minimizing the delay-margin function leads to determining the failure regions. Note that we choose this objective instead of merely maximizing the delay on the domain, since reaching the point of critical delay should be sufficient to observe frequent failures. The overall flow of BOIS consists of three steps as shown in Fig. 1:

1) Bayesian Optimization: We use the BO framework in order to create a *surrogate model* of the objective function, which is highly accurate at the border of the failure regions.
2) Shift ($\vec{\mu}_q$) optimization: Various failure regions are determined based on the created *surrogate model* and the best shift vector ($\vec{\mu}_q^*$) is chosen.
3) Failure rate estimation: Importance sampling is executed based on the shift vector ($\vec{\mu}_q^*$).

A. Bayesian Optimization

Bayesian Optimization (BO) [15] is a framework for global optimization of black-box functions on a compact domain X and can be classified as a response surface method [16]. A key property of BO is that it tries to minimize the number of evaluations of the objective function, rendering it ideal for the optimization of functions that are costly to evaluate, like the simulations of circuit delay for failure rate estimation. Here, we use BO to find the failure regions of an SRAM cell by minimizing $g(\vec{x})$ over the variation parameter space.

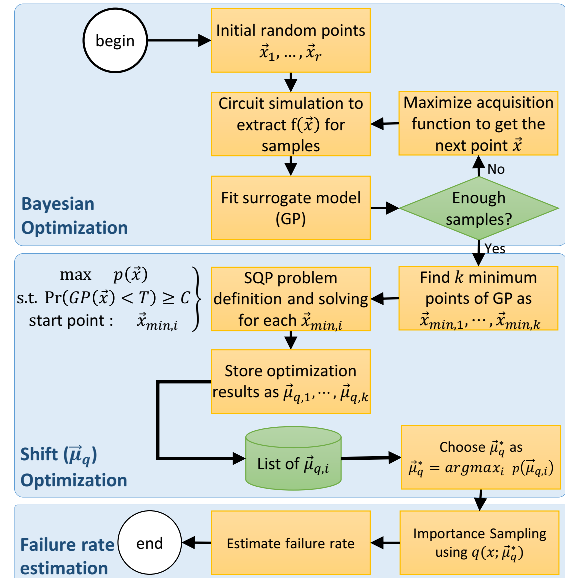

Fig. 1: Overall flow of BOIS methodology. The threshold T denotes the upper bound of the delay-margin below which a failure occurs with high probability.

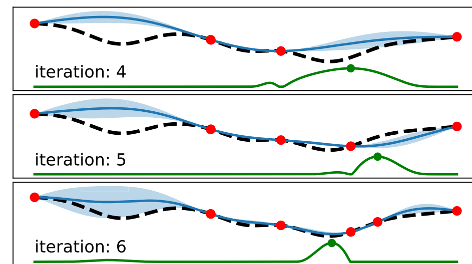

Fig. 2: Conceptual illustration of the *surrogate model* and the *acquisition function* for three iterations of BO. The dashed black line represents the unknown objective function which should be minimized. The red dots indicate past evaluations of $\vec{x}$ and $g(\vec{x})$, i.e. D. The *surrogate model* (GP) and its uncertainty (confidence region) are displayed in blue and light blue respectively. The *acquisition function* (green) with its maximum indicates the next point to evaluate.

Bayesian Optimization has two main ingredients, namely the probabilistic *surrogate model*, and the *acquisition function*. The *surrogate model* is used to express Bayesian belief about the response surface of the objective $g(\vec{x})$ given previously observed tuples of evaluations $(\vec{x}, g(\vec{x}))$, while the *acquisition function* is used to choose the next point $\vec{x}$ to evaluate.

The full procedure of BO, as presented in Fig. 2, is carried out as follows: Given an initial set of evaluations, which could be sampled at random, the belief about the response surface, i.e. the *surrogate model* is updated with respect to those evaluations. Then, the *acquisition function*, expressing the preference about points $\vec{x}$, is maximized with respect to the *surrogate model* to obtain the next iterate.

a) Gaussian Process as surrogate models: Gaussian Processes (GP) [17] are frequently used as *surrogate models* for

BO, since they can model uncertainty and offer closed-form expressions for the expected value and variance of their predictions. In other words, a GP can model a *distribution* of values of $g(\vec{x})$ for each $\vec{x}$ in the parameter space. For a specified GP and a set of data points $D = \{(\vec{x}_1,\ g(\vec{x}_1)), ..., (\vec{x}_n,\ g(\vec{x}_n))\}$, i.e. a set of tuples of design parameters $\vec{x}$ with their respective simulated delay-margin $g(\vec{x})$, the probability distribution of $g(\vec{x}^*)$ for an $\vec{x}^* \in X$ can be obtained by conditioning the GP on $\vec{x}^*$ and D. This can be interpreted as restricting the distribution of available functions to only those that agree with the observed data D.

We can thus use the GP as a probabilistic model for the delay-margin $g(\vec{x})$ of the SRAM cell given the simulations of delay-margins executed so far and extract the expected value and standard deviation of the estimated delay-margin for each given parameter configuration $\vec{x}$, namely $E_{GP}(\vec{x})$ and $\sigma_{GP}(\vec{x})$. We use the Radial Basis Function (RBF) kernel for the GP since it creates an infinitely smooth *surrogate model* and is suitable to reflect a physical phenomenon like SRAM delays.

b) Acquisition function: We use the *Probability of Improvement* (PI) *acquisition function* [18] for selecting the next data point $\vec{x}$. The PI can simply be understood as the probability of observing a better value of $g(\vec{x})$ given the *surrogate model*. Thus, for a given *surrogate model* $GP(\vec{x})$ of the delay-margin, the PI is defined as follows:

$$PI: \quad \alpha(\vec{x}) = Pr\big(GP(\vec{x}) < \tau\big),$$

where τ denotes the smallest value of delay-margin $g(\vec{x})$ observed so far. To find the next best point $\vec{x}$, this function is maximized and the point yielding the maximum value is chosen as the next query point. Although evaluations of the *acquisition function* are usually much cheaper than evaluations of the true objective function, the problem is generally non-convex and thus not easy to optimize. We therefore conduct several runs of the L-BFGS-B Algorithm [19] with random initial starting values and choose the best result as maximum.

c) BO for guided sampling of the response surface: Even though BO represents an optimization procedure, we only use it to obtain an approximation of the response surface of the delay-margin. To achieve this, we formulate an optimization problem to find design parameter configuration $\vec{x}$ with delays $f(\vec{x})$ close to critical delay value f_{crit} using (6), which should be solved by BO. Note that we choose this objective instead of merely minimizing the delay on the domain, since reaching the point of critical delay should be sufficient to observe frequent failures. Furthermore, those configurations $\vec{x}$ should not be too far from the origin, i.e. nominal delay with no variations.

As BO solves this problem, a probabilistic model of the delay-margin is created via the *surrogate model* (GP). Since the selection of data points used to generate the probabilistic model is guided by the optimization procedure (through the *acquisition function*), we expect more points in the region of lower objective values i.e. delay-margin (as illustrated in Fig. 2). This high point density should ensure a higher model accuracy for the low objective value region, i.e. failure region borders.

To minimize the delay-margin, we first generate an initial set of evaluations by sampling r points at random and simulate

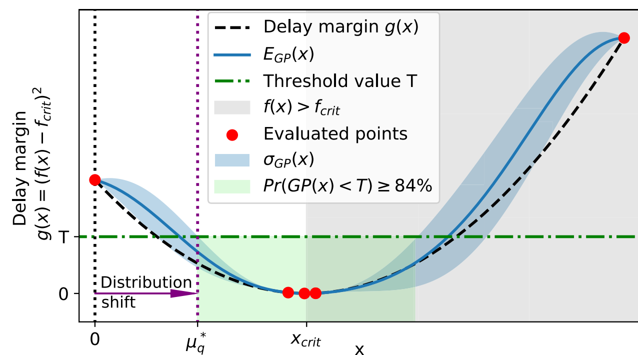

Fig. 3: A conceptual illustration of shift vector ($\vec{\mu}_q$) optimization problem given by (7).

their delay-margins. These points are then used as data D to initialize the *surrogate model* (GP). After initialization, BO is run for N steps using the PI *acquisition function*.

B. Shift ($\vec{\mu}_q$) optimization

Going back to our initial goal of achieving an accurate estimation of the failure rate P_f via IS, we now consider both factors of P_f, namely the weights $w(\vec{x})$ and the number of failures $\mathbb{1}(f(\vec{x}) > f_0)$. The weights $w(\vec{x})$ will be high for $q(\vec{x}; \vec{\mu}_q)$ close to $p(\vec{x})$, while the number of failures will be high in the region of high delay, which can be approximated using the *surrogate model* from BO.

A realistic circuit component such as an SRAM may have multiple failure regions. The failures are typically dominated by the failure region which is much closer to the origin, because $p(\vec{\mu}_q)$ for this region is much higher.

We determine the points belonging to the failure regions by selecting the BO points that have a high probability (C) of having a delay-margin under a certain threshold T according to the *surrogate model* (GP). Although the *surrogate model* already predicted a small delay-margin for these points, they may not have a high probability $p(\vec{x})$. Therefore, we try to locate a point with higher probability by solving the optimization problem defined as:

$$\max_{\vec{x}}\ p(\vec{x}) \quad \text{s.t.} \quad Pr(GP(\vec{x}) < T) \geq C. \tag{7}$$

The constraint will only consider points that have a delay-margin lower than T with a probability (confidence level) higher than C. The given problem is then solved using Sequential Quadratic Programming (SQP) [20]. The result of the optimization problem is a candidate shift vector $\vec{\mu}_q^*$, which may be used as the mean point of the sampling distribution $q(\vec{x}; \vec{\mu_q}^*)$. A conceptual illustration of the problem can be seen in Fig. 3.

IV. Experiments

A. Setup

Our approach is evaluated on a 6T-SRAM cell design (Fig. 4), implemented in Cadence Virtuoso environment using the 28nm FDSOI library. Circuit characteristics such as read latency time (RLT) are extracted via simulation using the Spectre Simulator. Variations are added to the transistor parameters, which are assumed to be normal distributed and mutually

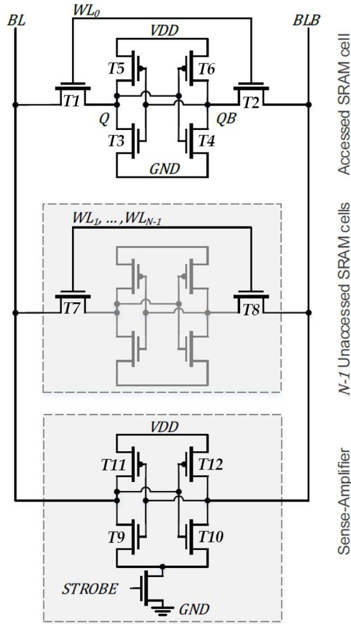

Fig. 4: 6T-SRAM and sense-amplifier in a memory column.

TABLE I: Two scenarios : Operation modes of 6T-SRAM cell

	VDD	f_{crit} in Delay-margin	#Transistors	Failure rate
Scenario 1	0.5V	10ns	12	Low Sigma
Scenario 2	1.0V	350ps	12	High Sigma

independent. Besides the 6T-SRAM cell, a sense amplifier was added to the circuit, which performs the read operation. Moreover, a virtual SRAM-cell was added, which accounts for the read disturb of the remaining cells in the column (here 255 cells). The access transistors of this representative cell have a 255 times wider channel width and the worst case is considered, where they store logical '0', while logical '1' is read from the target cell [12]. In total, we have 12 parameters with variation from the transistors of the 6T-SRAM cell, sense amplifier, and the virtual cell [12].

We compare our methodology with Gradient Importance Sampling [12], as it is shown to be superior compared to other related work. The comparison is done for two scenarios, Scenario 1 which represents the rare events, and Scenario 2 which represents the extremely rare events (high sigma). The operation mode parameters are presented in Table I.

For our experiments, we initialize the optimization procedure with $r = 5$ random points and continue to sample another $N = 195$ samples through BO to build the *surrogate model*. For the domain of $\vec{x} \in X \subset \mathbb{R}^d$, where d indicates the number of design parameters of the circuit, we choose $X = [-3\sigma, 3\sigma]^d$ for the low-sigma problem (Scenario 1) and $X = [-12\sigma, 12\sigma]^d$ for the high-sigma problem (Scenario 2). The value for σ is chosen as the a-priori standard deviation of $(\vec{x})_i, i \in \{1, .., d\}$. Regarding the Shift Optimization of μ_q, we use a small T relative to the f_{crit} value and a confidence level of $C = 84\%$ in our experiments.

Scenario 1: At the beginning, a golden MC with 10000 simulations was run using the conventional approach (no IS). Next, the MPFP ($\vec{\mu}_q$) was extracted using our proposed BOIS method as well as GIS [12]. The MPFP calculated from GIS was the local minima found by applying gradient-descent optimization according to GIS [12]. For both approaches, the MPFP search was terminated after 200 simulations. It is important to note, that the norm of the MPFP found by BOIS is smaller than the one found by GIS. This leads to higher weights $w(x)$ in (5), and thus faster convergence, as each sample drawn from $q(x)$ has higher significance.

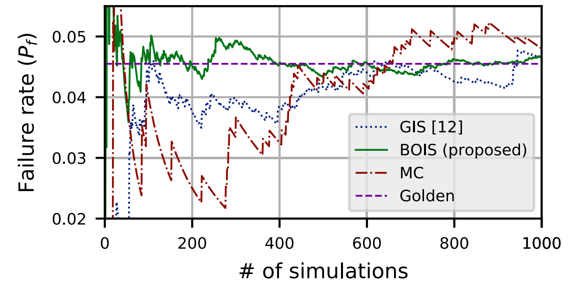

Fig. 5: Failure rate P_f estimates for Bayesian Optimization (BOIS), Gradient Importance Sampling (GIS) [12], Monte Carlo method (MC) and Golden result (Scenario 1).

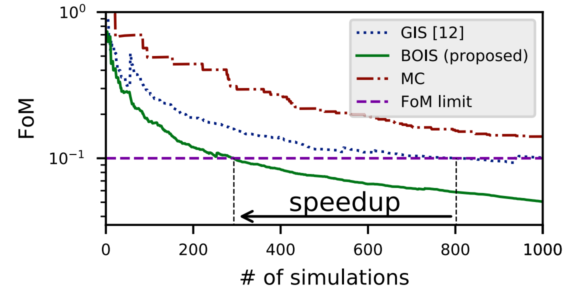

Fig. 6: Accuracy of IS estimate using different approaches - BOIS achieves 2x speedup compared to Gradient Importance Sampling (GIS) [12] for the same accuracy (Scenario 1).

After extraction of the MPFPs, 1000 Importance sampling runs were carried out for both approaches. The results of the failure rate estimates, depicted in Fig. 5, show that the BOIS approach converges faster to the golden result compared to GIS [12]. This fact can also be derived from Fig. 6, where the accuracy of the estimate is plotted against the number of simulations. The estimate is considered as accurate, when the FoM falls below 0.1, which ensures a relative error below 10% with confidence of 90%. A 2x speedup due to the reduction in required simulations is achieved by the proposed BOIS method compared to the GIS method [12]. As expected, the conventional MC simulation performed the worst, as they did not reach the 0.1 FoM margin within 1000 simulations at all. The reason why BOIS performed better is due to the fact that it sampled from a failure region closer to the origin (i.e. smaller MPFP norm) of the problem space. Table II presents the speedup of the proposed BOIS method and the breakdown of the number of simulation required for MPFP evaluation (BO) and IS steps.

Scenario 2: In this scenario, the probability of a failure of the 6T-SRAM cell is much smaller, and an accurate estimation becomes harder, or even unfeasible for the conventional MC methods. For instance, for a failure rate of 10^{-15}, the number of required MC simulations is $N_{MC} = 10^{17}$ [14], which is clearly infeasible. As the fail event of the SRAM cell is

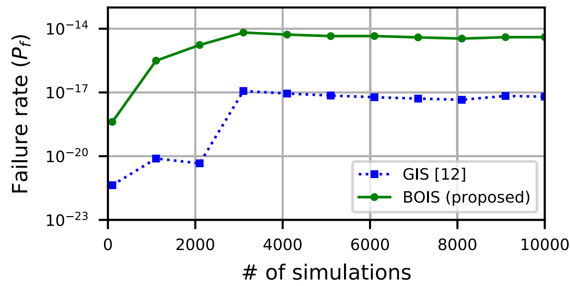

Fig. 7: Failure rate (P_f) estimates of Importance Sampling simulations for GIS [12] and BOIS at high-sigma (Scenario 2). The estimated values saturated for more than 3000 samples.

now much less likely, the IS runtime was increased and the algorithms terminated after 10000 simulations. Nevertheless, the number of simulations for finding the MPFP was again 200 as set in Scenario 1. The estimates of both approaches are compared in Fig. 7. In a normalized variation space (Section II-A), the norm of the MPFP ($\vec{\mu}_q$) for BOIS is 7.3 whereas the norm of the MPFP for GIS is 11. The large difference between the two MPFPs reveals that the GIS method [12] has overlooked a failure region closer to the origin, because the GIS method only follows the gradient to find the failure region without paying attention to other failure regions (see Fig. 8). Therefore, the MPFP found by GIS [12] led to an underestimation of the failure rate by about 632 times (Table II). In general, bad estimates are related to smaller failure rates, as the failure rate from which they are sampled have less significance, which is automatically expressed in (5), as the weights $w(x)$ become very small. This is a common performance criteria of IS algorithms and shows the superiority of our approach compared to GIS. A reason for GIS [12] finding a less efficient MPFP is its limitation for finding only local minima, in contrast to BOIS, which is capable of finding global minima, and thus better MPFPs.

V. Conclusion

In this work, we propose a novel Importance Sampling methodology, which is capable of estimating failures related to very rare events. Our algorithm is based on a Bayesian Optimization routine, which explores the problem space in a very efficient way and extracts the most probable failure regions. The advantages of this approach are its potential of global minimization and constant complexity irrespective of parameter dimensions. The simulation results, based on comparing our proposed method to the state-of-the-art approach on a 6T-SRAM cell, show that our proposed method reaches the same accuracy in failure rate estimation at least two times faster for rare events and 2 orders of magnitude

TABLE II: Comparison of GIS and BOIS (speedup, accuracy)

	FoM = 0.1 (speedup comparison)			
Scenario 1	Runs	BOIS	GIS	Speedup
	MPFP	200	200	-
	IS	293	802	2.7x
	Total	493	1002	2.0x
	10000 IS simulations (accuracy comparison)			
Scenario 2	Runs	BOIS	GIS	Ratio
	$\hat{P}_f$	$4.02 * 10^{-15}$	$6.36 * 10^{-18}$	632x

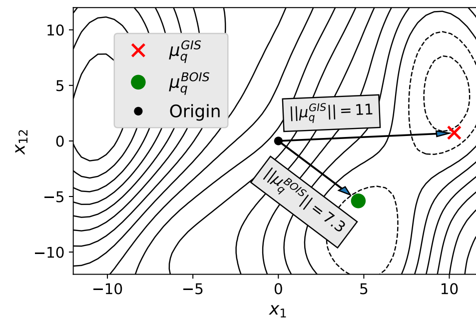

Fig. 8: The contours illustrate the delay-margin of Scenario 2 (high-sigma problem). The two dimensions x_1 and x_{12} are the normalized threshold voltages of transistors T1 and T12 in Fig. 4, respectively. These dimensions highlight the highest distance between μ_q^{GIS} and μ_q^{BOIS}.

higher failure rate for high sigma problems, as it considers multiple important failure regions.

References

[1] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, Nov 2005.

[2] L. Wilson, "International technology roadmap for semiconductors (ITRS)," *Semiconductor Industry Association*, 2013.

[3] J. Boley *et al.*, "Leveraging sensitivity analysis for fast, accurate estimation of sram dynamic write v min," in *DATE*, 2013.

[4] P. Weckx *et al.*, "Non-Monte-Carlo methodology for high-sigma simulations of circuits under workload-dependent BTI degradation – Application to 6T SRAM," in *IRPS*, 2014.

[5] A. Singhee and R. A. Rutenbar, "Statistical blockade: a novel method for very fast monte carlo simulation of rare circuit events, and its application," in *DATE*, 2007.

[6] ——, "Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design," *TCAD*, vol. 28, no. 8, pp. 1176–1189, 2009.

[7] Y. Zhao *et al.*, "Statistical rare event analysis using smart sampling and parameter guidance," in *System-on-Chip Conference (SOCC)*, 2015.

[8] W. Wu *et al.*, "Rescope: High-dimensional statistical circuit simulation towards full failure region coverage," in *DAC*, 2014.

[9] J. Zhai *et al.*, "An efficient Bayesian yield estimation method for high dimensional and high sigma SRAM circuits," in *DAC*, 2018.

[10] M. Wang *et al.*, "Efficient Bayesian yield optimization approach for analog and SRAM circuits," in *DAC*, 2017.

[11] ——, "Efficient Yield Optimization for Analog and SRAM Circuits via Gaussian Process Regression and Adaptive Yield Estimation," *TCAD*, 2017.

[12] T. Haine *et al.*, "Gradient importance sampling: An efficient statistical extraction methodology of high-sigma sram dynamic characteristics," in *DATE*, 2018.

[13] M. Wang *et al.*, "High-dimensional and multiple-failure-region importance sampling for SRAM yield analysis," *TVLSI*, vol. 25, no. 3, pp. 806–819, 2017.

[14] L. Dolecek *et al.*, "Breaking the simulation barrier: SRAM evaluation through norm minimization," in *ICCAD*, 2008.

[15] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*. Springer, 1975, pp. 400–404.

[16] S. Amaran *et al.*, "Simulation optimization: a review of algorithms and applications," *Annals of Operations Research*, vol. 240, no. 1, pp. 351–380, 2016.

[17] C. E. Rasmussen and C. K. Williams, *Gaussian process for machine learning*. MIT press, 2006.

[18] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," *Journal of Basic Engineering*, vol. 86, no. 1, pp. 97–106, 1964.

[19] R. H. Byrd *et al.*, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[20] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.