Hyperspherical Clustering and Sampling for Rare Event Analysis with Multiple Failure Region Coverage

Wei Wu
UCLA, EE Department
Los Angeles, CA
weiw@seas.ucla.edu

Srinivas Bodapati
Intel Corporation
Santa Clara, CA
srinivas.bodapati@intel.com

Lei He
UCLA, EE Department
Los Angeles, CA 90095
lhe@ee.ucla.edu

ABSTRACT

Statistical circuit simulation is exhibiting increasing importance for circuit design under process variations. It has been widely used throughout the design of standard cell circuits (SRAM, Flip-Flop, etc.) to maximize yield, i.e. to minimize the failure probability. Existing approaches cannot effectively analyze the failure probability when failed samples are distributed in multiple disjoint regions, nor handle the circuits with a large number of variations. To tackle these challenges, the proposed hyperspherical clustering and sampling (HSCS) approach first identifies multiple failure regions through a reweighted spherical k-means algorithm, which clusters failed samples on a set of hyperspheres, rather than the high dimensional open space. Next, a modified mixture importance sampling is designed to draw samples at those clusters to achieve multiple failure region coverage. The proposed HSCS is evaluated using both mathematical and circuit-based examples. It achieves about 3-order speedup over Monte Carlo with the same level of accuracy, while other importance sampling based approaches either fail to converge or converge to wrong results. Furthermore, HSCS demonstrates excellent robustness by generating consistent results in multiple replications.

CCS Concepts

•**Hardware** → **Process variations; Yield and cost modeling;**

Keywords

Process Variation, Yield, Clustering, Failure regions

1. INTRODUCTION

As integrated circuits (ICs) scale to smaller footprints than ever before, circuit reliability has become an area of growing concern due to the uncertainty introduced by process variations [1, 2, 3, 4, 5]. For highly duplicated standard cells, or critical circuit modules, such as PLLs, which stabilize the clock for the circuit system, an extremely rare failure event could lead to catastrophe of the entire chip.

ISPD'16, April 03-06, 2016, Santa Rosa, CA, USA

DOI: http://dx.doi.org/10.1145/2872334.2872360

In general, deterministic analysis of rare-event is infeasible [6]. Modern statistical circuit simulation approaches consider process variations and statistically simulate the circuit to estimate the probability that a circuit does not meet the performance metric. Among those approaches, Monte Carlo (MC) analysis remains the gold standard [7]. It repeatedly draws samples and evaluates circuit performance via transistor-level simulation. Even though the circuit simulation has been considerably accelerated [8, 9, 10], it is, however, extremely time-consuming because millions of samples need to be simulated to capture one single failure when the failure is a rare event.

Instead of sampling randomly with standard MC, more efficient approaches only sample the statistically likely-to-fail case [6, 11, 12, 13, 14, 15, 16, 17, 18]. These approaches, however, become less effective while analyzing circuits with a large number of variation parameters, or dealing with problems that failed samples are spread in multiple disjoint failure regions, which is becoming common in real circuit designs [16, 18, 19, 20].

(1) **Importance Sampling:** As a classic modification of MC, importance sampling (IS) modifies the MC sampling strategy. A critical step of IS is to construct a new "proposed" sampling distribution under which a rare event becomes less rare so more failures can be captured. Previous work investigated different approaches [11, 12, 13, 14]. For example, mixture importance sampling (MixIS) [11] mixes a uniform distribution, the original distribution, and a shifted distribution centered around the failure region. Method in [12] spherically searches the failure sample with minimal L2-norm (min-norm), then shifts the sample mean to the min-norm point. [14] shifts the sample mean to the centroid of the failure samples. All these approaches are related to mean shifting and assume that all failed samples are located in one region. However, in reality, failed samples may spread in multiple disjoint regions. In this scenario, the existing IS approaches cannot effectively cover all the failure regions, hence, leading to inaccurate and inefficient estimations.

(2) **Classification:** Approaches in this category tackle the problem from a totally different angle. As a representative example, statistical blockade (SB) [6] utilizes a classifier to block samples that are unlikely to fail, leaving only likely-to-fail samples to simulate. More recently, recursive SB [15] and REscope [16] are proposed to tackle problem with multiple failure regions. However, recursive SB assumes that each failure region is associate with different label, which does not hold for several circuits [19, 20]. Alternatively, REscope [16] relies on support vector machine (SVM) with radial basis function (RBF) kernel to identify failure regions, but SVM works as a black box model and is out of user's

control. Excessively training the SVM to identify multiple regions could easily lead to overfit.

Among others, [17, 21] uses a set of sample "chains" to explore the failure region with the aid of the Markov Chain Monte Carlo (MCMC) method. However, it is difficult to cover the entire failure region with several chains of MCMC samples, particularly when tens or hundreds random variables are considered. Multi-cone approach [18] deterministically breaks the original sample space into multiple non-overlapping cones, and sums up the analytically calculated failure probability in each cone. It does consider multiple failure regions, but the number of cones grows exponentially to the dimensionality, limiting it only effective for low dimensional problems.

In this paper, a hyperspherical clustering and sampling approach, HSCS in short, is proposed to effectively handle the challenges of both multiple failure regions and high dimensionality. As the first step, HSCS identifies multiple failure regions by grouping the failure samples into multiple clusters. Instead of clustering in a high dimensional open space, we sample spherically and develop a weighted spherical k-means algorithm to identify clusters only on a set of hyperspheres. Searching for min-norm points in these clusters is much easier than conventional spherical IS. Next, a modified mixture importance sampling shifts the sample mean to the min-norm points of multiple clusters so as to cover multiple failure regions.

HSCS is evaluated and compared with MC and other IS based implementations in terms of accuracy, efficiency, and robustness. On a small 2-dimensional problem with mathematically known distribution, HSCS yields very accurate results compared with mathematically calculated groundtruth. On a 70-dimensional charge pump circuit, HSCS is about 3 orders faster than MC and provides the same level of accuracy, while other IS based approaches either fail to converge or converge to wrong results. Furthermore, on both examples, HSCS demonstrates excellent robustness by generating consistent results in multiple replications.

The remainder of this paper is organized as follows. In Section 2, the rare event model problem and IS are reviewed. In Section 3, we expatiate the proposed algorithm, including spherical presampling and hyperspherical clustering step, and the modified MixIS step. In Section 4, HSCS is verified on a mathematically known distribution and a 70-dimensional charge pump circuit. This paper is concluded in section 5.

2. BACKGROUND

2.1 Rare Event Analysis

Let $f(X)$ be a probability density function (PDF) for a multivariate random variable X (e.g., a set of process variable parameters) which is the input of a measurement process as shown in (1); the output Y is an observation (e.g. memory read/write time, amplifier gain) with input X:

$$\underbrace{X}_{\text{variable}} \Rightarrow \boxed{\text{Measurement, SPICE, etc.}} \Rightarrow \underbrace{Y}_{\text{observation}} \quad (1)$$

In statistical circuit simulation, it is of great interest to estimate the probability of Y from a small subset $\mathcal{S}$ of the entire sampling space. For example, $\mathcal{S}$ can be the "failure region" for a circuit design and includes all samples that fail to meet the performance specification. Therefore, the probability $P(Y \in \mathcal{S})$ can be estimated as:

$$P(Y \in \mathcal{S}) = \int I(X) \cdot f(X) dX \quad (2)$$

where Y is the observation/performance with the input variable X and the indicator function $I(\cdot)$ outputs 1 only when $Y \in \mathcal{S}$, 0 otherwise. Note that the integral in equation (2) is intractable because $I(X)$ is unavailable in analytical form. Therefore, sampling based method must be used. For example, the MC method enumerates as many samples of X as possible (e.g., $x_1, \cdots, x_n$) according to $f(X)$ and evaluates their indicator function values to estimate $p(Y \in \mathcal{S})$ as:

$$\tilde{P}(Y \in \mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} I(x_i) \xrightarrow{\substack{a.s. \\ n \to +\infty}} P(Y \in \mathcal{S}). \quad (3)$$

Here $\tilde{P}(X \in \mathcal{S})$ is an unbiased estimate from sampling method and can be very close to $p(X \in \mathcal{S})$ with a large number of samples.

2.2 Importance Sampling

When $Y \in \mathcal{S}$ is a rare event, standard MC method becomes extremely inefficient. Importance sampling (IS) improves the efficiency by shifting the sample mean and sampling more statistically likely-to-fail (important) cases.

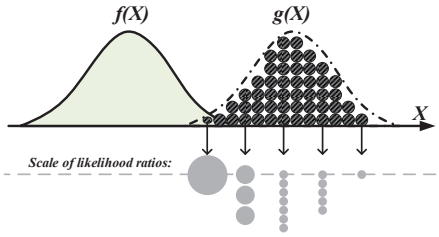

Figure 1: Likelihood ratios in mean-shift importance sampling

As illustrated using a 1-dimensional example in Figure 1, mean-shift IS samples from a proposed distribution $g(X)$ that tile towards $\mathcal{S}$ where a rare-event becomes less rare to happen:

$$P_{IS}(Y \in \mathcal{S}) = \int I(X) \cdot \frac{f(X)}{g(X)} \cdot g(X) dX \quad (4)$$

$$= \int I(X) \cdot w(X) \cdot g(X) dX. \quad (5)$$

Here, $w(X)$ is the likelihood ratio for each sample of X. $w(X)$ compensates for the discrepancy between $f(X)$ and $g(X)$ and unbiases the probability estimation under $g(X)$. Sampling based methods can be used to evaluate above integral as:

$$\tilde{P}_{IS}(Y \in \mathcal{S}) = \frac{1}{n} \sum_{j=1}^{n} w(\tilde{x}_j) \cdot I(\tilde{x}_j) \xrightarrow{\substack{a.s. \\ n \to +\infty}} P(Y \in \mathcal{S}). \quad (6)$$

where $\tilde{x}_j$ $(j = 1, \cdots, n)$ follows the distribution $g(X)$ rather than $f(X)$ because more likely-to-fail events can be sampled.

It is obvious that the samples closer to the nominal value are more desirable [12] because they are associated with greater probability $f(X)$ and likelihood ratio $w(X)$, hence have more significant impact on the estimated failure probability $\tilde{P}_{IS}$. In practice, most of the existing approaches shift the sample mean to the point that is closest to the origin on the accept/fail boundary, which is also known as the minimum-norm (min-norm) point [12]. However, the mean-shift IS implementations suffer from the following two drawbacks:

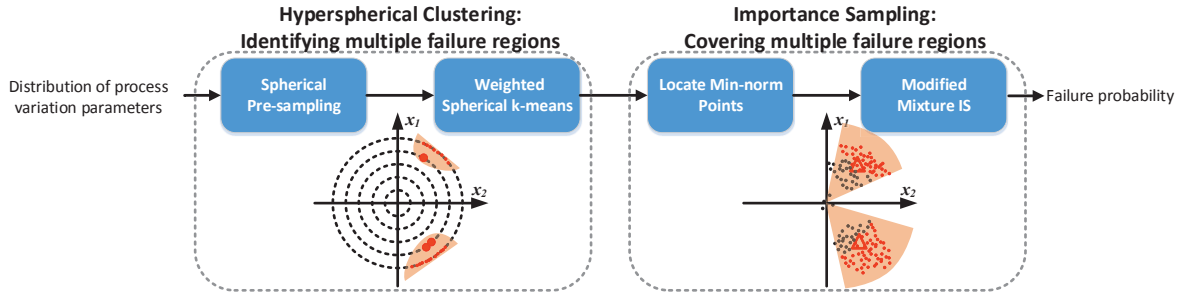

Figure 3: The HyperSpherical Clustering and Sampling (HSCS) algorithm consists of two phases: 1) hyperspherical clustering, 2) multiple mean-shift importance sampling.

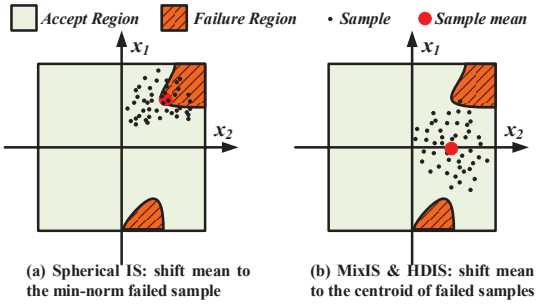

Figure 2: Existing mean-shift importance sampling implementations do not handle problems with multiple disjoint failure regions

First, they search the min-norm points by constructing an accept/fail boundary in the open space, which may take prohibitively long runtime, especially at high dimensionality.

Moreover, while existing approaches [11, 12, 13, 14] shift the sample mean to a more important point, they totally neglect that failed samples might be distributed in multiple disjoint regions. As illustrated in Figure 2, one shifted distribution might be insufficient to cover all the failures, hence leading to a biased estimation of $\tilde{P}_{IS}(Y \in \mathcal{S})$ in (6).

To improve the mean-shift IS, the remaining challenges turn out to be 1) identifying failure regions in high dimensional sample space, 2) effectively sampling to cover multiple failure regions.

3. HYPERSPHERICAL CLUSTERING AND SAMPLING

3.1 Algorithm Overview

In this section, we present the proposed hyperspherical clustering and sampling approach (HSCS). It consists of two major phases, (1) hyperspherical clustering, (2) importance sampling around multiple min-norm points, as illustrated in Figure 3. HSCS takes in the process variation parameters, and outputs the estimated failure probability $\tilde{P}_{IS}$ based on given requirements on performance metric Y.

To accurately estimate $\tilde{P}_{IS}$, we attempt to cover more samples that are closer to the nominal value. During the clustering phase, a weighted clustering algorithm is designed to bias the cluster centers towards those samples. In the second phase, sample means are shifted to the min-norm points of multiple clusters for two purposes: 1) capture more samples with greater weights, 2) cover the failed samples in multiple failure regions.

In the remaining part of this section, we will elaborate each phase of the algorithm.

3.2 Hyperspherical Clustering

The hyperspherical clustering phase includes a spherical presampling step and a weighted hyperspherical k-means step to cluster the failed samples. Algorithms in this phase are targeted to find the direction of failure regions, so that statistical approaches can be applied afterwards to estimate the failure probability with a better failure region coverage.

3.2.1 *Spherical Presampling*

In order to identify multiple failure regions, it is intuitive to collect a number of likely-to-fail samples (typically samples in the quantile of the performance distribution), and to cluster them into several aggregations according to their locations in the sample space. However, clustering samples that are randomly generated in high dimensional open space is challenging. Even in the same cluster, samples may still be far apart from each other. In this scenario, a cluster centroid does not necessarily mean more failed samples, leading to meaningless clusters.

Alternatively, we restrict the samples to a few hyperspherical surfaces by sampling spherically. In this scenario, clustering algorithms can be performed on a more restricted area rather than the high dimensional open space.

As illustrated in the left part of Figure 3, samples are randomly generated on hyperspheres with gradually increasing radius to capture samples in the quantile. During the implementation, we generate 1000 samples on each hypersphere surface and stop expanding the hypersphere until 5% or more samples on the current hypersphere surface fall in the 1% quantile.

3.2.2 *Weighted Hyperspherical K-means*

Conventional clustering algorithms (e.g. k-means) group samples to optimal clusters by minimizing the sum of Euclidean distance [22] between samples and their corresponding cluster centers, as defined in (7).

$$EuclideanDistance(X^{(1)}, X^{(2)}) = \left\| X^{(1)} - X^{(2)} \right\| \quad (7)$$

$$CosineDistance(X^{(1)}, X^{(2)}) = 1 - \frac{{X^{(1)}}^T X^{(2)}}{\|X^{(1)}\| \, \|X^{(2)}\|} \quad (8)$$

As we generate samples on hyperspheres, Euclidean distance makes less sense because the distance between samples and the origin is the same. It is more desirable to cluster samples based on the directions those samples pointing to rather than Euclidean distance. Therefore we use cosine

distance, defined in (8), as the distance metric, leading to a hyperspherical version of k-means algorithm.

Furthermore, a naive hyperspherical k-means algorithm only makes use of the samples on the outermost hypersphere, without incorporating the failed samples captured on the inner hyperspherical surfaces, which are usually associated with greater likelihood ratio according to (4), i.e. higher importance. To take full advantage of all the failed samples, we propose a weighted hyperspherical k-means algorithm. Each failed sample is normalized to unit length and associated with a weight calculated based on its probability density. With a targeted number of clusters k, the proposed algorithm returns the cluster assignment for each input failed sample.

Algorithm 1 Weighted Spherical K-Means Algorithm

Input: A set of M failed samples: $\mathcal{X} = \{X^{(1)}, X^{(2)}, ..., X^{(M)}\}$
Sample weights: $w^{(1)}, w^{(2)}, ..., w^{(M)}$
Number of initial clusters: k
Output: Cluster label for samples: $\mathcal{Y} = \{y^{(1)}, y^{(2)}, ..., y^{(M)}\}$
Updated number of clusters: k

1: Randomly initialize the unit length cluster centroids $\mathcal{U} = \{\mu^{(1)}, \mu^{(2)}, ..., \mu^{(k)}\}$;
2: **repeat**
3: **Cluster Assignment** (update $\mathcal{Y}$):
For each sample $X^{(i)}$, set $y^{(i)} = \underset{j}{\text{argmax}}\ {X^{(i)}}^T \mu^{(j)}$;
4: **Remove Empty Clusters** (update k)
Remove $\mathcal{X}_j$ if $\mathcal{X}_j = \{X^{(i)} | y^{(i)} = j\} = \emptyset$;
Update number of cluster k;
5: **Weighted Centroid Update** (update $\mathcal{U}$):
For cluster k, let $\mathcal{X}_j = \{X^{(i)} | y^{(i)} = j\}$, update centroid as $\mu^{(j)} = \sum_{X^{(i)} \in \mathcal{X}_j} w^{(i)} X^{(i)}$;
$\mu^{(j)} = \mu^{(j)} / \left\| \mu^{(j)} \right\|$;
6: **until** <$\mathcal{Y}$ remains unchanged>
7: Return $\mathcal{Y}$ and k;

As the first step of Algorithm 1, a set of initial cluster centroids are randomly generated. Next the algorithm iteratively updates the cluster label assigned for all samples, cleans up empty cluster, and recalculates the centroids, until the label assignment remains unchanged after one iteration. During the cluster assignment step, the algorithm checks the cosine distance between a sample and all cluster centroids. The cluster j that maximizes ${X^{(i)}}^T \mu^{(j)}$, which is equivalent to minimizing the cosine distance, will be selected. Moreover, we assign samples different weights in the centroid update process in step 5, therefore the centroids are biased to samples with higher importance.

One caveat is that k-means searches for the cluster assignment $\mathcal{Y}$ in a greedy fashion, resulting in convergence to the local optimal instead of guaranteeing global optimum. The proposed weighted hyperspherical k-means is not an exception. In practice, we start from multiple set of randomly initialized cluster centroids $\mathcal{U}$, and choose the one leading to minimal sum of cosine distance as the solution. Hence, the final solution could be more prone to take the global optimum.

Also, the number of clusters, k, is unknown before the clustering. In practice, we try a number of different k and choose the one with a trade off between the model complexity and goodness of fit. In the machine learning community, k is empirically chosen to be $\sqrt{M}$ [23], where M is the total number of samples to be clustered. More discussion on choosing k is included in the experiment section with concrete example.

3.3 Multiple Mean-Shift Importance Sampling

The previous phase generates normalized cluster centers, i.e. the direction of failure regions. In this phase, we locate the min-norm points of multiple failure regions and apply a modified mixture importance sampling (MixIS) approach to sample in all the failure regions and to estimate the overall failure probability.

3.3.1 *Locating the Min-norm Points using Bisection*

To locate the min-norm points more accurately and efficiently, we only search towards the direction of the clusters given that they have been identified.

Mathematically, all the samples in the same cluster can be covered by a cone defined in (9).

$$\mathcal{C} = \{X | CosiceDistance(X, \mu) \leq d_{max}\} \tag{9}$$

As illustrated in the right part of Figure 3, the opening angle of cone $\mathcal{C}$ is constrained by d_{max}, the largest cosine distance between failed samples in this cluster and the cluster centroid μ.

Algorithm 2 Locate min-norm points for each cluster with bisection

Input: Minimal radius of existing failure samples, R
Output: Radius of min-norm point: R_{min}

1: $R_{max} = R$;
2: $R_{min} = 0$;
3: **repeat**
4: $R = (R_{max} + R_{min})/2$;
5: simulate a small set of samples at Radius $= R$ in current cluster;
6: **if** any failed sample captured **then**
7: $R_{max} = R$;
8: **else**
9: $R_{min} = R$;
10: **end if**
11: **until** $R_{max} - R_{min} < R_{threshold}$
12: Return R;

Next, we apply bisection to search the minimal radius that leads to a failure in each cone, as presented in Algorithm 2. Starting with a lower bound of 0, and upper bound at the minimal radius of the existing failure samples, the algorithm bisects the radius and only simulates a small number of samples at this radius. It will reduce the upper bound to search the lower half region if any failure is captured during the simulation, otherwise, it will go to the upper half.

After locating the minimal radius R_i of a cone, the min-norm point of the corresponding cluster is calculated as $Cm_i = \mu_i * R_i$, where μ_i is the normalized cluster center that indicates the direction of this cluster.

3.3.2 *Modified Mixture Importance Sampling*

Next, we modifie the MixIS and shift the sample mean to all these min-norm points found in the previous step. The proposed distribution $g(x)$ is defined as

$$g(X) = \alpha f(X) + (1 - \alpha) \sum_{i=1}^{k} \beta_i f(X - Cm_i) \tag{10}$$

where $\beta_i = \frac{\sum_{X^{(i)} \in \mathcal{X}_k} w^{(i)}}{\sum_{\forall X} w^{(i)}}$ is the weight for each failure region (cluster), which is calculated based on the sum of sample weights in the cluster.

Note that we also keep a small ratio (α) of $f(x)$ in the proposed distribution $g(x)$, so that IS likelihood ratio

$$\frac{f(X)}{g(X)} = \frac{f(X)}{\alpha f(X) + (1-\alpha)\sum_{i=1}^{k} \beta_i f(X - Cm_i)} < \frac{1}{\alpha} \quad (11)$$

is bounded by $1/\alpha$. It prevents the likelihood ratio from going to infinity at certain X, and preserves the numerical stability of the modified MixIS.

4. EXPERIMENT RESULTS

The proposed HSCS is first evaluated using a mathematically known 2-dimensional normal distribution with 2 disjoint failure regions. Next, we verify HSCS using a more realistic high-dimensional charge pump circuit, which is known to have multiple failure regions.

4.1 Evaluation on Mathematically Known Distribution

On a sample space with 2-dimensional normal distribution, two disjoint failure regions, $\mathcal{S}_1$ and $\mathcal{S}_2$, are defined as follows:

- $\mathcal{S}_1 = \{X | \, \|X\| > 3.8 \text{ and } \phi(X) \in [\frac{2}{3}\pi, \frac{3}{4}\pi]\}$
- $\mathcal{S}_2 = \{X | \, \|X\| > 3.9 \text{ and } \phi(X) \in [\frac{4}{3}\pi, \frac{3}{2}\pi]\}$

where $\|X\|$ is the 2-norm of the sample, i.e. the Euclidean distance between the sample and the origin, and $\phi(X)$ is the phase of the 2-D sample. These two failure regions are illustrated in Figure 4[1].

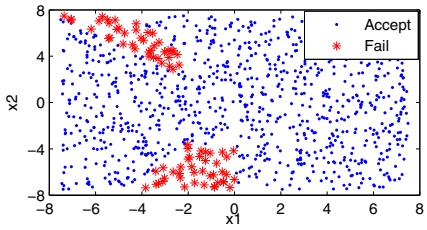

Figure 4: 2-dimensional sample space with two disjoint failure regions $\mathcal{S}_1$ and $\mathcal{S}_2$

Since the PDF and the failure regions are mathematically known, the failure probability can be calculated by integrating PDF function in (12),

$$P_F = \int_{X \in \{\mathcal{S}_1, \mathcal{S}_2\}} f(X)dX \approx 7.199e-5 \quad (12)$$

leading a failure probability of 7.199e-5, which is close to 4 sigma.

As the first step, HSCS gradually increases the radius of the sphere to search for failed samples and stops expanding until enough failed samples are collected. As illustrated in Figure 5, the presampling step converges at 4-sigma sphere in this particular example.

Obviously, those failed samples are aggregated in two separate regions in Figure 6(a). The weighted hyperspherical k-means updates the cluster assignments in a greedy fashion by always assigning a sample to its closest centroid. Hence, if the initial centroids are improperly selected, it is possible

[1]Figure 4 is plotted using uniformly distributed samples for better illustration.

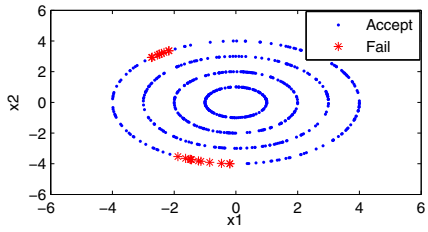

Figure 5: Spherical presampling to collect failed samples

that the iterative cluster assignments end up with assigning all samples in one cluster as illustrated in Figure 6(b), while leaving the other cluster empty (the empty cluster is removed in step 3 of Algorithm 1).

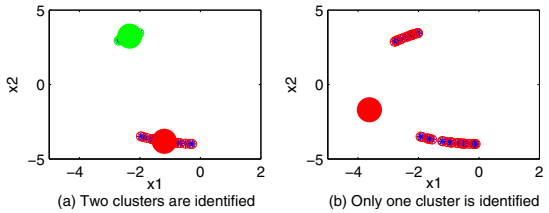

Figure 6: Spherical k-means might converge to local optimal with "improper" initial centroids

This problem has been well addressed in the machine learning community by randomly creating multiple set of initial centroids and applying the same cluster algorithm to all these set of samples. Only the cluster assignment with best optimization target, i.e. the smallest sum of cosine distance, will be chosen.

Next, bisection is applied to locate the min-norm points. In this example, we generate 20 samples only at each Radius. In each cluster, the algorithm ends up with 5 iterations and converges to radius at 3.9375 and 3.8125, which are very close to the groundtruth.

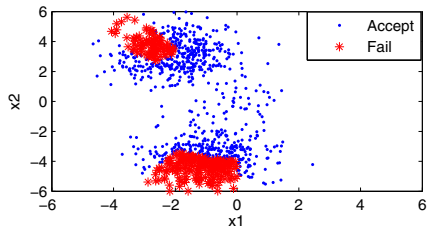

Figure 7: Sample coverage of the modified mixture importance sampling

The modified mixture importance sampling shifts the sample means to the min-norm points of both failure regions. Samples drew by importance sampling in Figure 7 indicate that both failure regions are accurately and fully covered.

The failure rate is estimated at 7.109e-5 using HSCS, which is quite close to the mathematically calculated ground truth, 7,199e-5. As a stochastic algorithm, we also run the HSCS with 100 replications to verify the stability. The estimated failure probability ranges from 5.54e-5 to 9.05e-5, with an average of 7.21e-5.

4.2 Experiments on Charge Pump Circuit

4.2.1 Charge Pump Circuit and Experiment Setting

We also evaluate the proposed HSCS using a charge pump (CP) circuit, which is a critical sub-circuit of the phase-locked loop (PLL), as illustrated in Figure 8. CP adjusts the frequency of the output clock signal, CLK_{out}, via a charge/discharge capacitor and voltage controlled oscillator (VCO).

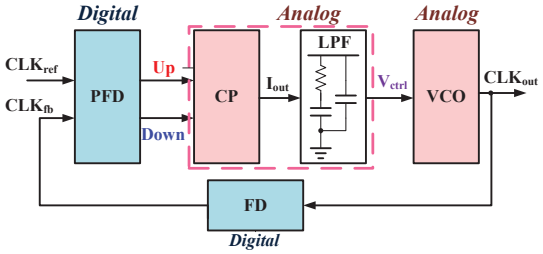

Figure 8: A block diagram of PLL

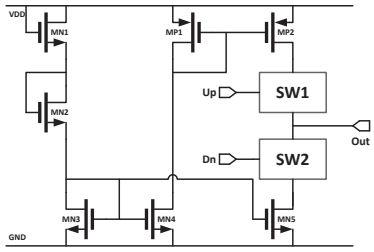

Figure 9: Simplified schematic of the charge pump

A simplified schematic of the charge pump consisting of two switched current sources is presented in Figure 9. Ideally, MN3, MN4, and MN5 on the bottom of Figure 9 are designed with the same dimension. The drain current flowing through these three NMOS transistors should be identical because they are imposed the same gate voltage. The same current also flows through MP1 since it shares the same branch with MN4. Similarly, on the top of Figure 9, two PMOS transistors form another current mirror, so that the current can be copied from MP1 to MP2. In this scenario, the charge current flowing through MP2 should be identical to the discharge current through MN5 when both switches are turned on, leading to zero net current.

In reality, it is, however, difficult to guarantee those transistors exactly the same dimension because of the process variation effects during chip fabrication. Mismatches on these transistors, especially on MP2 or MN5, could result in a nonzero net current at the output node. It could cause large fluctuation at the control voltage, also known as "jitter", which severely affect the PLL system stability. In the following experiments, we consider a failure if there is a big enough mismatch between the charge and discharge current, mathematically, $\max(\frac{I_{Charge}}{I_{Discharge}}, \frac{I_{Discharge}}{I_{Charge}}) > \gamma$, where γ is a threshold of this performance metric.

For experimental purpose, a CP circuit is designed using PTM 22nm high performance technology model [24] and simulated in HSPICE. The CP circuit is a typical circuit known to have multiple failure regions [19, 20, 16]. We analyze this circuit with two different process variation setups.

- In the first setup, we map the variations to threshold voltage (V_{th}), and only model the V_{th} of MP2 and MN5 as variation source. Hence, the failure regions can be visualized in a 2-dimensional space.

- A more comprehensive model with 10 parameters, including flat-band voltage (V_{fb}), threshold voltage (V_{th0}), gate oxide thickness (t_{ox}), mobility (μ_0), etc., are considered as variation source for each of those 7 transistors in Figure 9. Variations in those two digital switches are not accounted. In the second setup, there are a total of 70 variation parameters in the circuit, which is a relatively high dimensional problem.

In addition to the HSCS, Monte Carlo (MC) is included as the gold reference of the experiment. We also implement the high-dimensional importance sampling (HDIS) [14] and spherical importance sampling (SpIS) [12] for accuracy and efficiency comparison. The HDIS and Spherical IS are two typical mean shifting approaches that shift the sample mean to the centroid and min-norm point of the failure region respectively.

The efficiency is evaluated by counting the total number of simulations required to yield a stable failure rate. All the aforementioned approaches converge at the same criterion, i.e. the relative standard deviation of the estimated failure probability, $\sigma_r = \frac{std(p_f)}{p_f}$, gets smaller than 0.1.

4.2.2 2-D Setup with Visualized Failure Regions

In this setup, instead of investigating very rare failure event, we configure the threshold γ to target a 5% failure probability. Under this configuration, two failure regions can be easily visualized when we plot the accepted MC samples against failed ones in 2-dimensional sample space, as shown in Figure 10(a).

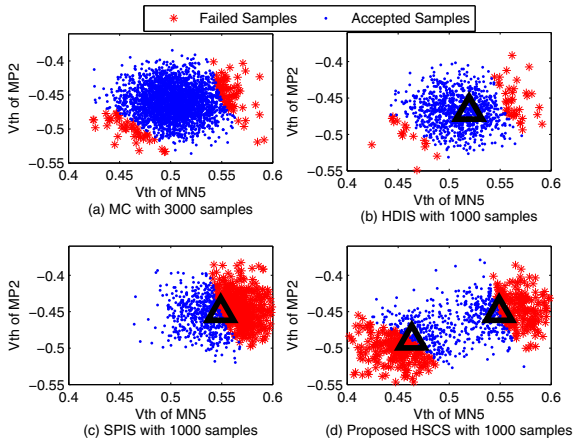

Figure 10: Multiple failure region coverage test MC, HDIS [14], Spherical IS [12], and HSCS

With only 1000 samples, the coverage of HDIS, Spherical IS, and the proposed HSCS are illustrated in Figure 10(b), (c), and (d), respectively. Sample means of these 3 importance sampling approaches are marked as upward-pointing triangulars in the Figures.

It is easy to notice that HDIS fails to shift the sample mean to any of the failure regions. As illustrated in 10(b), it attempts to draw samples around the centroid of the failed samples. The centroid of those failed samples, however, falls

almost close to the origin, which is obviously not in the failure region, leading to a poor coverage on those truly "important" samples.

Spherical IS shifts the sample mean to the existing sample with minimal norm. It correctly locate the min-norm point, as shown in Figure 10(c), but Spherical IS only samples one failure region while leaving the other one totally untouched.

The samples drew by the proposed HSCS are plotted in Figure 10(d). Samples generated during presampling and min-norm points searching are not included in this Figure. While the majority of samples are centered at the min-norm points of those two failure regions, HSCS still preserves a few samples around the origin to keep a small ratio of the original distribution according to equation (10) and avoids numerical instability in likelihood ratio calculation.

4.2.3 Hyperspherical Clustering with 70 Process Variation Parameters

In the following discussion, we model 10 process variation parameters on 7 transistors shown in Figure 9 in the CP circuit, leading to a 70-dimension problem. Transistors in two digital switches are not considered.

To collect enough samples for clustering, we generate 1000 samples at each hypersphere surface and gradually increase its radius and search for the samples on the 1% quantile. Until 6 sigma hypersphere, a total of M = 144 samples are collected, including 41 failed samples captured on 5 sigma hypersphere, and 103 failed samples on 6 sigma hypersphere. The weighted spherical k-means algorithm is applied on these 144 samples to group them into clusters. Note that the actual number of clusters (k_{actual}) generated by the algorithm could be small than k_{target}, as some clusters may become empty during the cluster assignment and are removed.

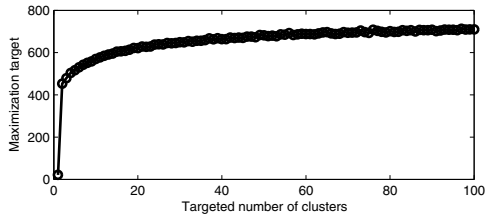

Figure 11: Clustering maximization objective while changing the targeted number of clusters

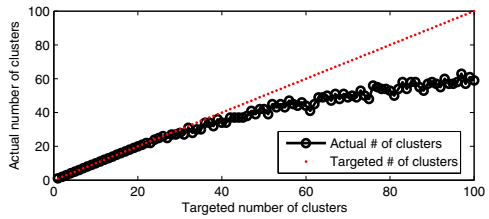

Figure 12: Number of actually clusters may be small than the targeted number of clusters

To determine the optimal number of clusters, we start with different k_{target} and evaluate the value of the maximization objective (also referred as profit) under those k_{target}. As shown in Figure 11, there is a big jump when k_{target} increases from 1 to 2. Afterwards, the slope becomes gentler and almost flat when k_{target} reaches 30.

A lot of information can be interpret from this Figure. First, the big jump indicates that the failed samples are located in two major clusters. When we use two centroids instead of one, the samples becomes much closer to the centroids, leading to a remarkable increase in the profit. Of course, these two big clusters can be further decomposed into smaller ones, but the profit generated by increasing k_{target} is smaller. When k_{target} is beyond 30, we do not benefit from increasing the cluster numbers.

The number of actually generated clusters k_{actual} is plotted against k_{target} in Figure 12, which helping us understand Figure 11 better. When k_{target} is small, the algorithm generates whatever number of clusters we ask for. Therefore, k_{actual} is overlapped with k_{target}. However, excessively increasing k_{target} results in a lot of redundant clusters, which are not assigned any samples and removed from the targeted clusters. These redundant clusters account for the gap between k_{actual} and k_{target}. In this particular problem, any k_{target} between 2 and 30 could be reasonable. As expected, the empirical guess, $k = \sqrt{M} = 12$ falls in this range.

4.2.4 Accuracy, Efficiency, and Robustness

The HSCS is also compared with MC, HDIS [14], and SpIS [12] in terms of both efficiency and accuracy. Their convergence curves are plotted in Figure 13[2], including one figure for the estimated failure probability (P_{fail}) and the other one for deviation of the estimation.

To generate the groundtruth, MC takes nearly 16 million simulations to get confident estimation of P_{fail} at 4.904e-5. The HDIS converges with only 4.9e4 samples (11k samples for pre-sampling and 38k for IS), but unfortunately, to a wrong estimation as shown in Figure 13(a). The Spherical IS is terminated since it does not show any sign of convergence after 7.4e5 samples being simulated. The poor performance of HDIS and SpIS is not a surprise because they fail to draw samples to comprehensively cover the failure regions, hence leading to fluctuant or event deviated estimations. More quantitative results of these approaches are presented in Table 1. Contrasting to HDIS and SpIS, the proposed HSCS achieves very promising estimation about 2.3e4 samples. In short, it estimates P_{fail} at MC accuracy with ~3 order speedup.

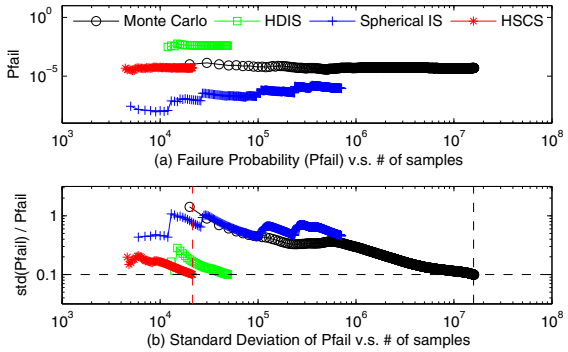

Figure 13: Convergence curve of Monte Carlo, HDIS, Spherical IS, and the proposed HSCS

[2]Note that the convergence curves of HDIS, SpIS, and HSCS start from different points because they need different # of samples in the presampling step.

Table 1: Accuracy and efficiency evalution on 70-dimensional charge pump circuit

	Monte Carlo	HDIS [14]	SpIS [12]	Proposed HSCS with 10 replications
failure probability	**4.904e-5**	**3.9e-3**	**8.788e-7**	3.89e-5 ∼ 5.88e-5 (mean **4.82e-5**)
Total #sim. runs	**1.584e7**	**3.8e4**	**>7.4e5**	4.6e3 ∼ 5.5e4 (mean **2.3e4**)
#sim. for presampling	-	1.1e4	4e3	4.2e3
#sim. for IS		3.8e4	>7e5	410 ∼ 5.1e4 (mean 1.9e4)

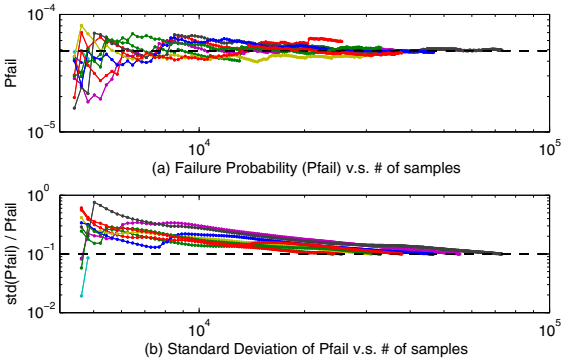

Figure 14: Robustness test of HSCS with 10 replications

To ensure that HSCS can consistently generate accurate estimation, we executed the same program with 10 replications and presented their convergence curves in Figure 14. We notice that the failure probabilities estimated by these replications converge to the ground truth, the dashline in Figure 14(a). As detailed in Table 1, the estimated failure probability ranges from 3.89e-5 to 5.88e-8, with an average of 4.82e-5. This is very close to the MC result. Also, it only takes an average of 2.3e4 samples to converge the simulation, which is about 3 orders faster than MC.

5. CONCLUSION

In this paper, HSCS is proposed to tackle the challenging statistical circuit simulation problems with multiple failure regions and high dimensionality, which are the shortcomings of the existing importance sampling and classification based approaches. HSCS first applies spherical presampling and clustering to identify multiple failure regions. Next, it locates the min-norm points of each failure region and leverage a modified MixIS that shifts the sample mean to those min-norm points. Therefore, the importance samples cover multiple failure regions. In the experiments on a 70-dimensional charge pump circuit, HSCS achieves ∼3 orders speedup over MC providing the same level of accuracy, while other IS based approaches either fail to converge or converge to wrong results. Furthermore, HSCS demonstrates excellent robustness by generating consistent results in multiple replications.

6. REFERENCES

[1] S. R. Nassif, "Design for variability in DSM technologies [deep submicron technologies]," in *ISQED*, 2000, pp. 451–454.

[2] K. Agarwal and S. Nassif, "The impact of random device variation on SRAM cell stability in sub-90-nm cmos technologies," *IEEE Trans. on VLSI Systems,*, vol. 16, no. 1, pp. 86–97, 2008.

[3] S. Wang, A. Pan, C. O. Chui, and P. Gupta, "Proceed: A pareto optimization-based circuit-level evaluator for emerging devices," in *ASP-DAC*. IEEE, 2014, pp. 818–824.

[4] G. Leung, S. Wang, A. Pan, P. Gupta, and C. O. Chui, "An evaluation framework for nanotransfer printing-based feature-level heterogeneous integration in vlsi circuits," 2015.

[5] S. Wang, H. Lee, F. Ebrahimi, P. K. Amiri, K. L. Wang, and P. Gupta, "Comparative evaluation of spin-transfer-torque and magnetoelectric random access memory," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, 2016.

[6] 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*, 2008, pp. 235–251.

[7] C. Jacoboni and P. Lugli, *The Monte Carlo method for semiconductor device simulation*. Springer, 1989, vol. 3.

[8] X. Chen, W. Wu, Y. Wang, H. Yu, and H. Yang, "An escheduler-based data dependence analysis and task scheduling for parallel circuit simulation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 702 –706, oct. 2011.

[9] W. Wu, Y. Shan, X. Chen, Y. Wang, and H. Yang, "Fpga accelerated parallel sparse matrix factorization for circuit simulations," in *Reconfigurable Computing: Architectures, Tools and Applications*. Springer, 2011, pp. 302–315.

[10] W. Wu, F. Gong, R. Krishnan, L. He, and H. Yu, "Exploiting parallelism by data dependency elimination: A case study of circuit simulation algorithms," *Design Test, IEEE*, vol. 30, no. 1, pp. 26–35, Feb 2013.

[11] 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 *Proceedings of the 43rd DAC*, 2006, pp. 69–72.

[12] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: SRAM evaluation through norm minimization," in *ICCAD*, 2008, pp. 322–329.

[13] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato, "Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis," in *ICCAD*, 2010.

[14] W. Wu, F. Gong, G. Chen, and L. He, "A fast and provably bounded failure analysis of memory circuits in high dimensions," in *19th ASP-DAC*, 2014, pp. 424–429.

[15] A. Singhee, J. Wang, B. H. Calhoun, and R. A. Rutenbar, "Recursive statistical blockade: an enhanced technique for rare event simulation with application to sram circuit design," in *21st Intl. Conf. on VLSI Design*. IEEE, 2008, pp. 131–136.

[16] W. Wu, W. Xu, R. Krishnan, Y.-L. Chen, and L. He, "REscope: High-dimensional statistical circuit simulation towards full failure region coverage," in *Proceedings of the 51st DAC*, 2014.

[17] C. Dong and X. Li, "Efficient SRAM failure rate prediction via gibbs sampling," in *Proceedings of the 48th DAC*, 2011.

[18] R. Kanj, R. Joshi, Z. Li, J. Hayes, and S. Nassif, "Yield estimation via multi-cones," in *Proceedings of the 49th DAC*, 2012.

[19] P. Mukherjee, C. S. Amin, and P. Li, "Approximate property checking of mixed-signal circuits," in *Proceedings of the 51st DAC*, 2014.

[20] P. Mukherjee and P. Li, "Leveraging pre-silicon data to diagnose out-of-specification failures in mixed-signal circuits," in *Proceedings of the 51st DAC*, 2014.

[21] S. Sun and X. Li, "Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space," in *ICCAD*, 2014, pp. 324–331.

[22] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[23] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*. Academic press, 1979.

[24] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20nm finfet design with predictive technology models," in *Proceedings of the 49th DAC*, 2012.