On Efficient Monte Carlo-Based Statistical Static Timing Analysis of Digital Circuits

Javid Jaffari[a,b] and Mohab Anis[a,b]
[a]Spry Design Automation, Waterloo, ON, Canada N2J 4P9
[b]ECE Department, University of Waterloo, Waterloo, ON, Canada N2L 3G1
{jjaffari,manis}@vlsi.uwaterloo.ca

***Abstract*—The Monte-Carlo (MC) technique is a well-known solution for statistical analysis. In contrast to probabilistic (non-Monte Carlo) Statistical Static Timing Analysis (SSTA) techniques, which are typically derived from simple statistical or timing models, the MC-based SSTA technique encompasses complicated timing and process variation models. However, a precise analysis that involves a traditional MC-based technique requires many timing simulation runs (1000s). In this paper, the behavior of the critical delay of digital circuits is investigated by using a Legendre polynomial-based ANOVA decomposition. The analysis verifies that the variance of the critical delay is mainly due to the pairwise interactions among the Principal Components (PCs) of the process parameters. Based on this fact, recent progress on the MC-based SSTA, through Latin Hypercube Sampling (LHS), is also studied. It is shown that this technique is prone to inefficient critical delay variance and quantile estimating. Inspired by the decomposition observations, an efficient algorithm is proposed which produces optimally low L_2-discrepancy Quasi-MC (QMC) samples which significantly improve the precision of critical delay statistical estimations, compared with that of the MC, LHS, and traditional QMC techniques.**

I. Introduction

A reliable Statistical Static Timing Analysis (SSTA) is pivotal in predicting the variabilities in digital VLSI circuits and addressing the variabilities in the design phases. Recently, several *probabilistic*-based (non-Monte Carlo) SSTA techniques have been proposed, where the signal arrival times are treated as random variables, and the Probability Distribution Functions (PDFs) of the circuit critical delays are achieved by proper statistical analysis. Blaauw et al. [1] provide a recent survey on SSTA techniques. The drawback of the current probabilistic SSTA techniques is that, each is based on models, where some of the timing and process variation effects are ignored or simplified. Such effects include, the nonlinearity of gate delays as a function of the process parameters and capacitive loads; the nonlinearity of arrival times due to max operations, causing non-zero skew signal arrival times; the interdependency among input/output rise/fall time and gate delay; interconnect delay models; non-Gaussian process parameters; and spatial/structural correlations. Therefore, the Monte-Carlo (MC) technique has recently attracted attentions as a suitable candidate for a reliable and accurate timing sign-off [2], because the MC technique can generally account for any complicated models by accepting the excessive runtime costs. Moreover, the development and integration costs of MC techniques are minimum, since the available deterministic-STA engines can be maximally reused in developing new MC-based SSTA tools. These are in addition to the benefits of simply breaking any MC implementation into parallel processes to reduce the overall runtime.

However, the problem of the traditional MC-based statistical analysis technique is its slow convergence rate ($O(N^{-1/2})$). Therefore, to achieve reasonably precise estimations of the statistical moments of the critical delay in timing sign-off, thousands of samples/simulations are required. The precision of an estimation is defined in terms of the confidence interval range in which the actual parameter of interest lies. For example, the 99% confidence interval of an estimator $\hat{\theta}$ is $\left[\mu_{\hat{\theta}} - 2.576\sigma_{\hat{\theta}}, \mu_{\hat{\theta}} + 2.576\sigma_{\hat{\theta}}\right]$. The standard deviation of the estimation, $\sigma_{\hat{\theta}}$, can be obtained by repeating the experiments with new random sample sets. It should be noted that the objective of this paper is to reduce the runtime of the MC-based SSTA by reducing the number of samples by improving the precision of the estimations.

Critical Aware Latin Hypercube Sampling (CALHS [3]), a recent study, whose focus is to improve the MC-based SSTA precision by Latin Hypercube Sampling (LHS) and stratification, have problems such as scalability and no significant advantage over the traditional MC technique for the critical delay variance estimation. The Quasi Monte Carlo (QMC) is another alternative to the MC technique by providing a faster convergence rate for low dimensional problems [4]. However, it will be shown and studied later that no runtime gain can be achieved even by using the traditionally generated QMC samples for timing variance and quantile estimation.

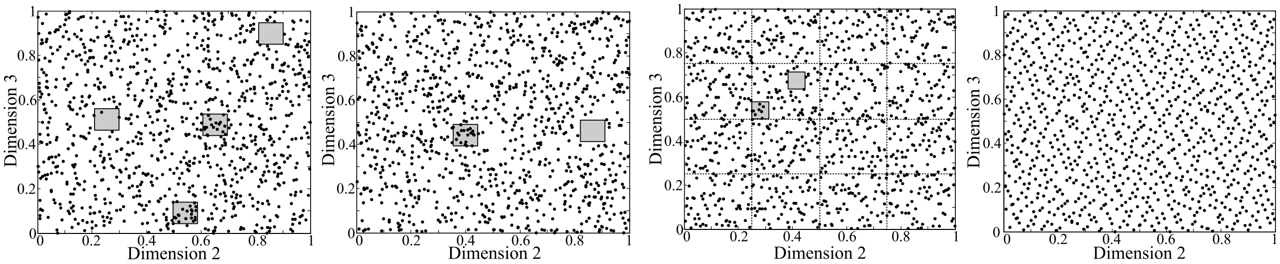
The preliminaries for the MC, LHS, and QMC are presented in Section II. In Section III, the behavior of critical delay, as a function of the process parameters, is quantitatively analyzed by using a Legendre polynomial-based ANOVA decomposition. Strong bivariate monomial terms in the decomposed function of the critical delay variance is observed, which supports the inefficiency of the CALHS and traditional QMC. Based on this observation, an algorithm is proposed in Section IV to generate optimally low L_2-discrepancy QMC sequences that significantly improve the precision of estimations over those of the MC, CALHS, and traditional QMC. Lastly, in Section V, a complete SSTA framework is proposed by combining the proposed enhance-precision QMC with the LHS.

II. Background

In this section, the MC-based statistical timing analysis is reviewed. Two advanced MC-based techniques, namely, LHS and QMC are also investigated.

The MC is known as a powerful tool for the numerical estimation of high-dimensional integrals. Therefore, the MC technique can also be utilized for statistical estimations in SSTA. Suppose, $\boldsymbol{p} = \left\{p^{(1)}, p^{(2)}, \ldots, p^{(d)}\right\}$ is a set of d-dimensional process parameters with a known Joint Probability Distribution Function (JPDF), $\phi_d(\boldsymbol{p}) : \Re^d \rightarrow \Re$. Each $p^{(i)}$ represents a process parameter, including, gate length, oxide thickness, RDF-driven threshold voltage, and interconnect dimension variations. If $D(\boldsymbol{p})$ is the critical delay of a circuit as a function of the

978-1-4244-2820-5/08/$25.00 ©2008 IEEE



(a) Monte Carlo Sampling (b) Latin Hypercube Sampling (c) CALHS [3], 4×4 stratified LHS (d) QMC Sampling (Sobol)

Fig. 1. 2-D projections of different sampling approaches. The gray squares represent areas with high or low concentration of samples.

process parameters, then the m-th statistical moment of D is formulated as the d-dimensional integral,

$$E\left[D^m\left(\boldsymbol{p}\right)\right] = \int_{\Re^d} D^m\left(\boldsymbol{p}\right)\phi\left(\boldsymbol{p}\right)d\boldsymbol{p}. \tag{1}$$

For the MC technique, a set of independently distributed uniform pseudo-random vectors are generated in the $[0,1]$ interval. Then, the set is converted through a simulation technique (e.g., the inverse transformation method) to samples of random variables with a given JPDF [5]. The samples are then used to calculate the critical delay (D) statistical moments through several calls of a deterministic-STA tool. The following is the MC estimation of Eq. (1) by using N samples:

$$\hat{E}_N\left[D^m\left(\boldsymbol{p}\right)\right] = \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) \quad , \qquad x_i \in [0,1]^d \tag{2}$$

where $x_1, x_2, \ldots, x_N$ are independent and identically distributed uniform d-dimensional vectors in $[0,1]^d$, and $f\left(\boldsymbol{x}\right) = D^m\left(\Phi^{-1}\left(\boldsymbol{x}\right)\right)$, where $\Phi^{-1} : [0,1]^d \to \Re^d$ is the inverse transformation function which generates samples with the JPDF of ϕ (JCDF of Φ) from the uniform and i.i.d $\boldsymbol{x}$.

However, each run of an MC-based SSTA with a new set of pseudo-random values can lead to a different estimate for $E\left[D^m\left(\boldsymbol{p}\right)\right]$ with an error of $e = E\left[D^m\left(\boldsymbol{p}\right)\right] - \hat{E}_N\left[D^m\left(\boldsymbol{p}\right)\right]$. The standard deviation of this error defines the probabilistic confidence interval range of the estimator. The greater the number of samples (N) is, the smaller the range is. The expected convergence rate of the MC technique is $O\left(N^{-1/2}\right)$. This indicates that to reduce the initial confidence range by ϵ, the number of samples should be increased by ϵ^2 times.

However, an important feature of the estimation error, e, is that it is related to the equi-distribution (uniformity) of the samples in $[0,1]^d$ rather than their randomness. This idea strongly suggests that by using a well-spread sequence, which is more uniform than a pseudo-random sequence, a more precise estimation can be achieved [4]. LHS [6] is a variance reduction technique which increases the convergence rate by providing more uniform samples in 1-D. This is achieved by partitioning the $[0,1]$ range into equal length subranges and generating the same number of samples in each subrange randomly. A random permutation of the LHS samples are finally adopted to generate the sample vectors. The uniformity of the LHS samples does not differ from that of MC technique in projections higher than 1-D. In the next section, it is demonstrated how this property impacts the LHS-based SSTA. Figure 1(b) shows the 2-D projection of the LHS-based samples. It can be seen that the samples are not much more uniform than the traditional MC-based samples (Fig. 1(a)).

CALHS, an LHS-based SSTA approach, is proposed in [3] and relies on criticality directed stratification to obtain a lower estimation error. Even though, the use of CALHS is to increase the higher dimensional unit hypercube uniformity by stratification, the uniformity is still limited due to the finite number of regions in each dimension ($r = 4$) and ignoring many other non-stratified projections. This is demonstrated in Fig. 1(c), where the 2-D projection uniformity is somewhat improved, but this is the only projection which has a higher uniformity. If more projections needed to be stratified then the number of strata increases non-polynomially (r^s, if s is the number of PCs selected for stratification).

Instead of generating random samples by a pseudo-random number generator, which is the base of both the traditional MC and LHS techniques, the QMC is the technique to produce deterministic low-discrepancy sequences which are more uniformly distributed over the problem space than the random samples. Higher than 1-D uniformity is apparent for such sequences, that leads to a faster convergence rate than that of the MC technique. The convergence rate of the QMC technique is $O\left(log^d N/N\right)$ which converge asymptotically faster than the MC [4]. It is concluded that the rate is no more superior than that of the MC, unless $N > e^d$, which is absolutely impractical for even moderate problems ($d > 10$). However, in Section III, it is explained why this is not so in practice, and how the QMC technique exhibits significant advantages over the MC technique for some types of high-dimensional problems (e.g., high-dimensional computational finance problems [7]). Figure 1(d) depicts the 2-D projection of the QMC samples, generated by the Sobol algorithm [8]. A very high uniformity is observed in this projection. There are other algorithms to generate low discrepancy sequences such as the Halton [9], Faur [10], and Niederreiter [11].

Before closing this section, the quantitative measure of the discrepancy in a 2-D projection is reviewed. $L_2\left(X\right)$, the L_2-discrepancy of X, provides a measure of uniformity for N samples X, as follows [12]:

$$L_2\left(X\right) = \sqrt{\begin{array}{l} \frac{1}{N^2}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\prod\limits_{k=1}^{2}\left(1 - \max\left(x_i^{(k)}, x_j^{(k)}\right)\right) \\ -\frac{1}{2N}\sum\limits_{i=1}^{N}\prod\limits_{k=1}^{2}\left(1 - {x_i^{(k)}}^2\right) + 3^{-2} \end{array}} \tag{3}$$

where $x_i^{(k)}$ is the k-th dimension of the i-th sample of X. The lower $L_2\left(X\right)$ is, the more uniform the distribution of the samples is. For example, the L_2-discrepancy of the samples, depicted in Fig. 1(a) to 1(d) are 115×10^{-4}, 88.3×10^{-4}, 38.2×10^{-4}, and 9.07×10^{-4}, respectively.

III. Legendre Polynomial-Based ANOVA Decomposition of Critical Delay

As mentioned in Section II, the LHS is inefficient for the estimation of the critical delay variance and quantile point. Moreover, it is mentioned that the QMC technique is surprisingly efficient for high-dimensional computational finance problems. In this section, the notion of the effective dimension is reviewed, and a numerical technique is used to quantify the effective dimension of the critical delay of a digital circuit by the Legendre polynomial-based decomposition. Finally, a discussion of the analysis is provided to describe the reason for the weakness of the LHS technique in SSTA, and to provide suggestions to improve the quality of the QMC results for the variance and quantile estimation in the Section IV.

The QMC technique's success in computation of some high-dimensional finance problems [7] was unexpected, given the Koksma-Hlawka error bound of $O\left(log^d N/N\right)$. The convergence rate for such problems is roughly $O\left(n^{-1}\right)$, independent of the problem dimension. Several researchers have attempted to explain this surprisingly good performance [13], [14]. A qualitative explanation, applicable for any general problem, is developed under the notation of effective dimension [15], [16]. The idea is that the integrand function $f\left(\boldsymbol{x}\right)$, defined in $[0,1]^d$, can be decomposed into a sum of orthogonal functions over the subsets of the problem variables. If a large portion of the total function variance comes from a few random variables or orthogonal functions with small dimensions, then the effective dimension is significantly lower than the nominal problem dimension, leading to more accurate results when using the QMC.

Consequently, by using the ANalysis Of VAriance (ANOVA) representation, $f\left(\boldsymbol{x}\right)$ can be decomposed into a sum of orthogonal functions of all the subsets of $\boldsymbol{x}$, as follows:

$$\begin{aligned} &f\left(\boldsymbol{x}\right) = \sum_{u\subseteq\ell} f_u\left(\boldsymbol{x}\right) = f_0 + \sum_{i=1}^{d} f_i\left(x^{(i)}\right) + \\ &\sum_{i<j}\sum f_{ij}\left(x^{(i)}, x^{(j)}\right) + \cdots + f_{1\cdots d}\left(x^{(1)}, \cdots, x^{(d)}\right) \end{aligned} \tag{4}$$

where $\ell = \{1, 2, \cdots, \mathrm{d}\}$. The ANOVA terms are orthogonal:

$$\begin{aligned} &\int_{[0,1]^d} f_u\left(\boldsymbol{x}\right) f_v\left(\boldsymbol{x}\right) d\boldsymbol{x} = 0 && \text{when u} \neq \text{v} \\ &\int_0^1 f_u\left(\boldsymbol{x}\right) dx^{(j)} = 0 && \text{for j} \in \text{u}. \end{aligned} \tag{5}$$

Therefore, if the variances of the integrand and the ANOVA terms are defined as

$$\begin{aligned} &\sigma^2\left(f\right) = \int_{[0,1]^d} f^2\left(\boldsymbol{x}\right) d\boldsymbol{x} - \left[\int_{[0,1]^d} f\left(\boldsymbol{x}\right) d\boldsymbol{x}\right]^2 \\ &\sigma^2\left(f_u\right) = \int_{[0,1]^d} f_u^2\left(\boldsymbol{x}\right) d\boldsymbol{x} \end{aligned} \tag{6}$$

the variance of the integrand function can be expressed as the sum of the variances of all the orthogonal functions, as follows:

$$\sigma^2\left(f\right) = \sum_{u\subseteq\ell} \sigma^2\left(f_u\right) \tag{7}$$

Consequently, the following two types of effective dimensions are introduced in [15]:

1) The effective dimension of f in the superposition sense is d_S, if $\sum_{|u|\leq d_S} \sigma^2\left(f_u\right) \geq p\sigma^2\left(f\right)$
2) The effective dimension of f in the truncation sense is d_T, if $\sum_{u\subseteq\{1,2,\cdots,\mathrm{d_T}\}} \sigma^2\left(f_u\right) \geq p\sigma^2\left(f\right)$

where p is a proportion chosen to be less than, but close to 1. For example if 99% of the variance of f is due to the components of $\boldsymbol{x}$ selected one at a time, the effective dimension in the superposition sense (d_S) is 1. This means the interactions among the parameters have a negligible effect on the function. Similarly, if $d_S = m$, then a large portion of the integrand variation is due to interactions of orders lower than m. Finally, the truncation sense of effective dimension is relate to a list of important variables. Therefore, $d_T = m$ means that the first m variables make up most of the integrand value.

The reasons for high quality of the QMC-based estimations due to effective dimension concept are given in [16], [17]. The efficiency is due to the fact that the low discrepancy sequences produce a high uniformity in the first few dimensions ($\leq$ 12) or low order projections ($\leq$ 3). Therefore, if most of the function variance comes from some few variables or low order interactions of all the variables, the QMC provides a better estimation than the MC-based techniques.

In this section, a numerical technique, first reported in [18], [19], is used to estimate the effective dimension of the statistical mean and standard deviation of a digital circuit's critical delay. To perform the estimation, the technique utilizes shifted Legendre polynomial functions as orthogonal function bases for the purpose of the integrand decomposition. Shifted Legendre polynomials are orthogonal in the $[0,1]$ range [19]. Therefore, $f\left(\boldsymbol{x}\right)$ is decomposed by

$$f\left(\boldsymbol{x}\right) = \sum_{r_1=0}^{\infty} \cdots \sum_{r_d=0}^{\infty} c_{\boldsymbol{r}} \prod_{j=1}^{d} \phi_{r_j}\left(x^{(j)}\right) \tag{8}$$

where $\phi_n(x) = \left[\int_0^1 p_n^2\left(2x-1\right) dx\right]^{-0.5} p_n\left(2x-1\right)$ is the n-th order shifted and scaled Legendre polynomial, if $p_n\left(x\right)$ is the n-th Legendre polynomial, and $c_{\boldsymbol{r}}$ is the constant coefficient for the combination of $\boldsymbol{r} = \left(r_1, \cdots, r_d\right)$ which is calculated as follows:

$$c_{\boldsymbol{r}} = \int_{[0,1]^d} f\left(\boldsymbol{x}\right) \prod_{j=1}^{d} \phi_{r_j}\left(x^{(j)}\right). \tag{9}$$

The unbiased estimator for $c_{\boldsymbol{r}}$ is estimated by the MC technique as follows [18]:

$$c_{\boldsymbol{r}}^2 = \frac{1}{N\left(N-1\right)} \left(\begin{array}{c} \left(\sum\limits_{k=1}^{N} f\left(x_k\right) \prod\limits_{j=1}^{d} \phi_{r_j}\left(x_k^{(j)}\right)\right)^2 \\ -\sum\limits_{k=1}^{N} f^2\left(x_k\right) \prod\limits_{j=1}^{d} \phi_{r_j}^2\left(x_k^{(j)}\right) \end{array} \right) \tag{10}$$

Finally, the effective dimensions of the integrand are estimated by setting $\sigma^2\left(f_u\right) = \sum\limits_{r\in R(u)} c_{\boldsymbol{r}}^2$, where

$$R\left(u\right) = \left\{ \left\{r_1, \cdots, r_d\right\} \left| \begin{array}{l} r_j \in Z, 0 \leq r_j \leq o \\ r_j = 0 \leftrightarrow j \notin u \\ \sum\limits_{j=1}^{d} r_j \leq m \end{array} \right. \right\} \tag{11}$$

where m and o are the maximum degree and order of the basis functions.

By using this numerical method, the effective dimensions of the ISCAS85 benchmark circuits critical delays are estimated and found to be one in the superposition sense. This occurs because the process parameters with spatial correlations such as gate length (L_g) contribute the most to the variance of the delay of long paths, which are most likely to be critical. The parameters produce additive-form functions of the singular monomials for the critical delays. In fact, the spatial statistically-correlated process parameters are decomposed into linear additive form of independent PCs by PCA [20]. Therefore, the delay of a gate, and finally, of a path has strong additive singular terms considering strong linear dependence between gate delay and process parameters. This idea results in one of the advanced probabilistic SSTA techniques in [20]. The analysis, described in this paper, indicates that for the sample C6288 benchmark, and for a maximum of the seventh order ($o = 7$), $\sum_{|u|=1} \sigma^2 (f_u) = 0.995\sigma^2 (f)$ for the critical delay mean, suggesting that the mean is effectively 1-D in superposition sense. However, in computing the standard deviation effective dimension, only 1% of the total $\sigma^2 (f)$ is found to be due to 1-D Legendre terms, wheras more than 98% is due to the 2-D polynomials. The reason for this is that when the standard deviation of an additive function of singular monomials is computed, the square of that function shows a strong pairing of monomials, leading to an additive function of bivariate monomials. This confirms that the effective dimension of the variance estimation is two in superposition sense.

It is now possible to predict that the LHS-based technique can provide a precise estimation of the critical delay mean as LHS-based samples are highly uniform in 1-D. However, no significant improvement in the standard deviation estimation can be achieved theoretically by the LHS-based SSTA since they do not provide a high 2-D uniformity. This conclusion is supported by the CALHS simulation results, presented in Section VI.

One other important conclusion is derived from the fact that the critical delay standard deviation is effectively 2-D. That is, to employ any QMC sequence for the SSTA effectively, it must be examined closely in terms of its L_2 discrepancy (2-D uniformity), and possibly improved in that sense.

IV. Proposed Algorithm for Low L_2-Discrepancy Sobol Sequences Suitable for SSTA

As seen in Section III, to efficiently estimate the mean and variance of a critical delay, a sampling technique is required that provides a high uniformity in at least 2-D projections. The QMC sequences are appropriate choices for this purpose. The Sobol [8] is a low discrepancy QMC sequence which is preferred over many other QMC sequences [9]–[11], especially for high-dimensional problems, due to faster generation, easier implementation, and a higher uniformity for both 1-D and 2-D projections [21]. However, due to the finite number of samples, even in the Sobol sequence, many 2-D projections show a high discrepancy, which is undesirable for an efficient SSTA analysis. Figure 2 illustrates some bad 2-D pairings for 1023 Sobol samples. It is evident that some regions of the 2-D projections are entirely empty. In contrast, an example of a good pairing is depicted in Fig. 1(d).

In this section, an algorithm is proposed which modifies the traditional Sobol algorithm to generate sequences with as many good pairings as possible, that is suitable for the SSTA. It is noteworthy that the high-dimensional finance problems have significant 1-D portions [18]. As a result, the QMC technique performs fairly well with no need to further optimize the Sobol sequence to generate better 2-D projections. Moreover, it is incorrectly assumed that it is not possible to predict a poor pairing, prior to the generation a complete sequence [21]. However, it is shown in this paper that, the bad pairing of the dimensions can, in fact, be detected in advance. Finally, two algorithm are proposed to detect the bad pairings and to generate the optimum Sobol sequences with minimum L_2-discrepancies.

Before this, a brief description of the Sobol sequence generation algorithm is given in the next subsection to show how a Sobol sequence can be improved in terms of its L_2-discrepancies.

A. Generating a Sobol Sequence

The Sobol sequence generation algorithm [8] is briefly reviewed now. To generate N samples of a d-dimensional Sobol sequence, $x_i^{(j)}$, where $i = 1, \cdots, N$ and $j = 1, \cdots, d$, each $x_i^{(j)}$ can be generated from the following:

$$x_i^{(j)} = a_1 v_1^{(j)} \oplus a_2 v_2^{(j)} \oplus \cdots \oplus a_W v_W^{(j)} \tag{12}$$

where $\oplus$ denotes a bitwise XOR operation, $v_k^{(j)}$ are direction numbers, and the $a_i \in \{0, 1\}$ coefficients are extracted from the binary representation of the Gray code of i, $G(i) = \sum_{k=0}^{W} a_k 2^k$. The Gray code of i is defined as $G(i) = i \oplus \text{int}\left[\frac{i}{2}\right]$, where $\text{int}[x]$ represents the largest integer inferior or equal to x. Thus, $W = [\log_2 i]$.

For example, to find $x_{25}^{(j)}$, the following steps are taken:

$$\begin{array}{l} i = 25 \rightarrow G(i) = 11001 \oplus 01100 = 10101 \\ \text{and hence,} \quad x_{25}^{(j)} = v_1^{(j)} \oplus v_3^{(j)} \oplus v_5^{(j)} \end{array} \tag{13}$$

where each direction number, $v_k^{(j)}$, is a binary fraction that is written as

$$v_k^{(j)} = m_k^{(j)} / 2^k \tag{14}$$

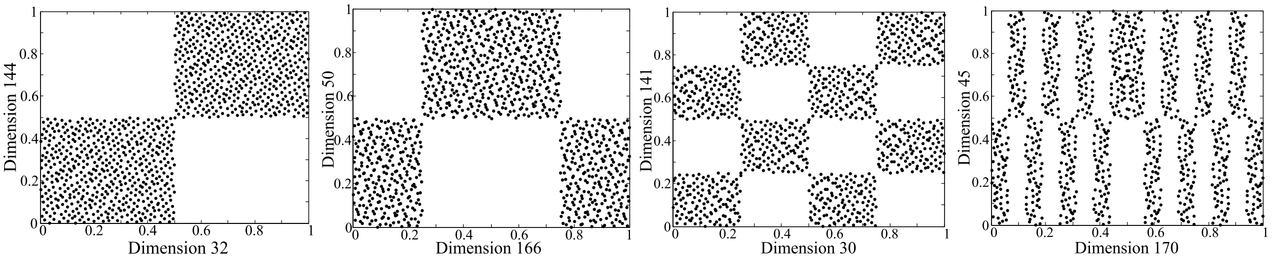
where $m_k^{(j)}$ is an odd integer, $0 < m_k^{(j)} < 2^k$ for $k = 1, \cdots, W$. For each dimension j, a sequence of integers $m_k^{(j)}$ is defined by a q-term recurrence relation as

$$m_i^{(j)} = \begin{array}{l} 2b_1^{(j)} m_{i-1}^{(j)} \oplus 2^2 b_2^{(j)} m_{i-2}^{(j)} \oplus \cdots \\ \oplus 2^{q-1} b_{q-1}^{(j)} m_{i-q+1}^{(j)} \oplus \left(2^q m_{i-q}^{(j)} \oplus m_{i-q}^{(j)}\right) \end{array} \tag{15}$$

where $b_k^{(j)} \in \{0, 1\}$, $k = 1, \cdots, q - 1$ are the coefficients of a q-degree primitive polynomial [22] specified for each dimension j. Jaeckel [23] offers a CD containing more than 8 million primitive polynomials up to degree 27 to be used for the Sobol generation.

It is evident in each dimension that there is a great deal of flexibility in choosing the initial values $(m_1^{(j)}, \cdots, m_q^{(j)})$, whereas the remaining $(m_{q+1}^{(j)}, \cdots, m_W^{(j)})$ is generated through the q-degree recurrence relation of Eq. (15). The constraints on the initial values $m_k^{(j)}$ for $k = 1, \cdots, q$ are that they must be odd integers and less than 2^k; therefore, for a dimension with a q-degree primitive polynomial, there are $2^{q(q-1)/2}$ possible choices in selecting the initial values. Consequently, a random technique is traditionally used to choose the initial $m_k^{(j)}$ terms for each dimension in [23].

By referring back to Fig. 2, to fill the empty regions and increase the uniformity of the samples, either more samples are needed or the initial values of the corresponding dimension should be changed. This is where the newly developed technique enters to picture. As a result, the objective of this part of the work is to pick a set of initial values which reduces the bad pairings as much as possible. Sobol, himself, has realized the importance of the initial values on the quality of the generated sequences, and proposed two properties to increase the uniformity of the samples [24]. However, to satisfy Sobol's proposed



(a) $L_2 = 8.02e-2$ (b) $L_2 = 4.04e-2$ (c) $L_2 = 2.07e-2$ (d) $L_2 = 1.06e-2$

Fig. 2. Some bad (high L_2-discrepancy) pairing of Sobol samples

properties, 2^{2d} samples are needed that is not practical for even moderate dimensional problems. Also, the property does not have anything to do with 2-D uniformity [21], [25]. Cheng and Druzdzel have defined a measure of 2-D uniformity and proposed a search algorithm to find a set of initial values with a high defined uniformity [26]. The drawback to their technique is that the number of samples and dimensions must be known in advance. Moreover, their technique re-produce Sobol sequences and re-evaluate their defined discrepancy measure in each iteration (after an initial value update) substantially increasing the runtime for large number of samples and dimensions. This is due to the incorrect assumption that poor dimension pairings cannot be found prior to the generation of sequences [21].

B. Optimizing the Initial Values to Maximize the Uniformity

To perform any optimization of the initial values, it is critical that the algorithm which is used to determine the L_2-discrepancy can generate the estimation efficiently. Otherwise, even though the process of finding the optimum initial values is a one-time task, it will not be tractable for large number of samples and dimensions. The naive implementation of Eq. (3) is $O\left(N^2\right)$, and the fastest implementation of that requires $O(N\left(\log N\right)^2)$ operations [27]. This is in addition to the need for regenerating N Sobol samples each time an initial value is updated.

However, we show that not only there is no need to generate Sobol sequences for L_2 calculation but also there is even no need to calculate L_2 from Eq. (3) to detect the bad pairings of dimensions. This task can be performed by defining some Boolean rules over the binary representation of the direction values. In another words, the bad pairings can be foreseen in advance by boolean investigating of initial values from the first point. Let's define the following notation: suppose $v_{i,b}^{(j)}$ is the b-th most-significant bit of the binary representation of $v_i^{(j)}$, the i-th direction value of the j-th dimension. And $x_{s,b}^{(j)}$ is the b-th MSB of the binary representation of $x_s^{(j)}$, the s-th Sobol sample of the j-th dimension. Therefore, from Eq. (12), the following relation is derived:

$$x_{s,b}^{(j)} = a_1 v_{1,b}^{(j)} \oplus a_2 v_{2,b}^{(j)} \oplus \cdots \oplus a_W v_{W,b}^{(j)} \tag{16}$$

From this simple relation, several Boolean rules are derived to predict the way the Sobol samples cover each projection. For example, a simplest rule is:

$$\begin{aligned} &\text{if } \forall i = 1, \cdots, W \quad v_{i,1}^{(d_1)} = v_{i,1}^{(d_2)} \\ &\Rightarrow \forall s = 1, \cdots, 2^W - 1 \quad x_{i,1}^{(d_1)} = x_{i,1}^{(d_2)} \end{aligned} \tag{17}$$

which means up to the ($2^W - 1$)-th sample, the projection of the d_1-th and d_2-th dimensions is similar to that in Fig. 2(a), since, $x_i^{(d_1)} < 0.5 \Leftrightarrow x_i^{(d_2)} < 0.5$. It is evident how fast such bad pairing can be detected.

More complicated rules can be achieved by using the bitwise XOR operation. For example, a pattern, similar to the one in Fig. 2(b) is generated, if $\forall i = 1, \cdots, W \quad v_{i,1}^{(d_1)} \oplus v_{i,2}^{(d_1)} = v_{i,1}^{(d_2)}$. Fortunately, a generic rule can be derived for such binary rules, for as high as the degree which is required. Moreover, a range for the L_2-discrepancy of the samples can be achieved for each rule. Here, is the general form of such binary rules and its corresponding L_2-discrepancy (L_2),

$$\begin{aligned} &\text{if } \forall i = 1, \cdots, W \\ &v_{i,r}^{(d_1)} \oplus \sum_{u \in l_m} v_{i,u}^{(d_1)} \oplus v_{i,m}^{(d_1)} = v_{i,r}^{(d_2)} \oplus \sum_{\nu \in l_n} v_{i,\nu}^{(d_2)} \oplus v_{i,n}^{(d_2)} \\ &\Rightarrow \text{for first } 2^{\mathrm{W}} - 1 \text{ samples :} \\ &0.08 \times 2^{(1-m-n)} > L_2\left(X, d_1, d_2\right) > 0.08 \times 2^{(2-m-n)} \end{aligned} \tag{18}$$

where $\sum$ is a bitwise XOR operator, and $l_m \subseteq \{r+1, \cdots, m-1\}$. The L_2 value reduces to half when m or n increases by one. This occurs because the L_2-discrepancy relates to the area of the largest rectangle with a constant number of samples, in the projection [4]. For example the areas of such rectangles are 0.5, 0.25, 0.125, and 0.0625 in Fig. 2, respectively, as the their L_2 values reduce with the same rate.

Algorithm 1 and 2 are developed according to the defined general Boolean rule of (18) to extract the lower bound of the L_2-discrepancy.

Finally, a simulation annealing optimization engine is developed which minimizes the number of bad pairings by switching the appropriate bits of the initial values. For a given W, the objective of the optimizer is to limit the maximum L_2-discrepancy of the pairs of dimensions $d = \{1, \cdots, 2^{W-MaxW}\}$, less than $0.08/2^{MaxW-1}$, where $MaxW = 1, \cdots, W-1$. Therefore, any pairing with $\{2^{W-MaxW-1}, \cdots, 2^{W-MaxW}\}$ dimensions from the lower dimensions should only be verified to satisfy as much as the $L_2 < 0.08/2^{MaxW-1}$ condition, speeding the L_2-discrepancy computation process. Finally, to make the optimizer converge faster, it is only the first $MaxW$ bits of the initial values in the dimensions of ($d = \{2^{W-MaxW-1}, \cdots, 2^{W-MaxW}\}$) which are included in the search during the optimization. Moreover, the simulation annealing engine is directed by an initial value selection criterion, giving high priority to those dimensions that have the worst discrepancies.

Figure 3 reflects the distribution of the calculated L_2-discrepancies (base on Eq. (3)) before and after the initial values are optimized. As depicted in Fig. 3(d), even for the first few dimensions $(1, \cdots, 32)$ before optimization, some pairs have

Algorithm 1 Calculate $L_2\left(MaxW\right)$

```
for $cnt = 1$ to $MaxW$ do
  if $cnt$ is even then
    $mx \Leftarrow (cnt/2) - 1$
  else
    $mx \Leftarrow (cnt - 1)/2$
  end if
  if $mx < 0$ then
    $mx \Leftarrow 0$
  end if
  $s_1 \Leftarrow$ non-empty subsets of set: $\{0, \cdots, mx\}$
  for $i = 1$ to $N\left(s_1\right)$ do
    $w_1 \Leftarrow s_1(i)$
    $m \Leftarrow$ last element of $w_1$
    $s_2 \Leftarrow$ all subsets of set: $\{w_1(1) + 1, \cdots, cnt - m - 2\}$
    for $j = 1$ to $N\left(s_2\right)$ do
      $w_2 \Leftarrow s_2(j)$
      $ne \Leftarrow N\left(w_2\right)$
      for $k = ne$ down to $1$ do
        $w_2(k + 1) \Leftarrow w_2(k)$
      end for
      $w_2(1) \Leftarrow w_1(1)$
      $w_2(ne + 2) \Leftarrow cnt - m - 1$
      if (CheckRule($w_1, w_2$) or CheckRule($w_2, w_1$)) is true
      then
        return $0.08/2^{(cnt-1)}$
      end if
    end for
  end for
end for
return $0$
```

Algorithm 2 CheckRule(w_1, w_2)

```
for $i = 1$ to $W$ do
  $d_1xor \Leftarrow$ false
  for $j = 1$ to $N\left(w_1\right)$ do
    $d_1xor \Leftarrow d_1xor \oplus v_{i,w_1(j)+1}^{(d_1)}$
  end for
  $d_2xor \Leftarrow$ false
  for $j = 1$ to $N\left(w_2\right)$ do
    $d_2xor \Leftarrow d_2xor \oplus v_{i,w_2(j)+1}^{(d_2)}$
  end for
  if $d_1xor \neq d_2xor$ then
    return false
  end if
end for
return true
```

very high discrepancies $(L_2 > 0.08)$ and many others have discrepancies higher than the maximum of the optimized version $(L_2 > 0.01)$. However, as shown in Fig. 3(e)-3(h) for the optimized version, the maximum discrepancy reduces to half in each step from 256 dimensions down to 32.

V. The Proposed SSTA

The proposed SSTA framework is established by combining low discrepancy Sobol sequences and the LHS technique. The number of each type is related to the total number number of samples. For a given number of samples $N = 2^W - 1$, 2^{W-1} dimensions use Sobol samples, whereas the reminder dimensions use LHS samples.

Algorithm 3 Optimize Initial Values (W)

```
Generate initial IVs
Compute $L_2$s for all pairs
Initialize priorities of IVs based on $L_2$ values
while there is a bad pairing ($L_2 > 0$) do
  while inner-loop criterion do
    Randomly select an IV, directed by priorities
    Switch the value of an appropriate bit
    Update the altered pairings' $L_2$s
    if accept(Pairing cost, Temperature) then
      Apply the changed bit to the selected IV
    end if
  end while
  Update Temperature
end while
```

The optimum initial values of the Sobol generator for a given W is pre-computed and stored by using the algorithm, proposed in Section IV. Since the Sobol samples provide low 1-D and 2-D discrepancies, they are prioritized for assigning them to PCs of the process parameters. As discussed in Section III, the PCs contribute the most to the variance of critical delay. The LHS samples are used to provide samples for the non-spatially correlated process parameters (e.g., RDF) or any remaining PCs to provide an efficient mean estimation.

The number of Sobol dimensions is limited (2^{W-1}) for a given number of samples. However approaching the first dimension, the 2-D uniformities increase. Therefore, it is beneficial to order the PCs, so that the most important components, which contribute more to the circuit critical delay, use the lower discrepancy dimensions. Consequently, a weight is assigned for each PC as a measure of its criticality. The following is used to derive the criticality of each PC:

$$c_i = \sum_{j=1}^{p} \psi_{i,j} \sum_{k=1}^{N_j} \exp\left\{\alpha \cdot \left(\frac{Slack_{j,k}}{D_{nom}}\right)^2\right\} \tag{19}$$

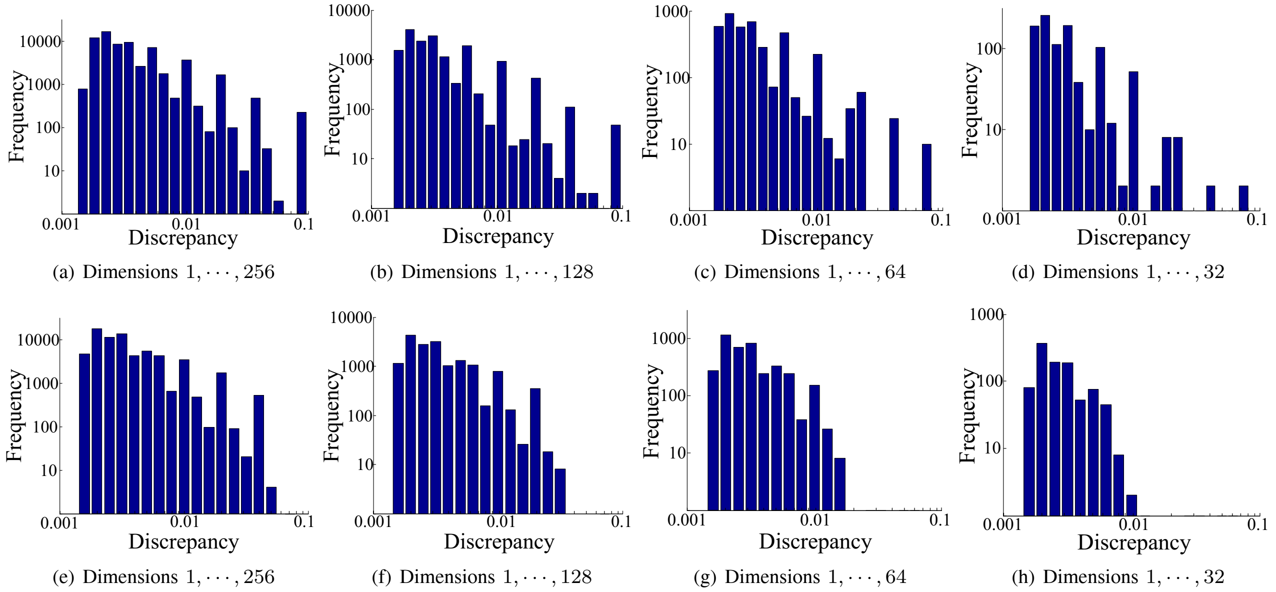
where c_i is the measure of the criticality of the i-th principal component, p is the number of PCs, $\psi_{i,j}$ is the coefficient of the j-th PC in the i-th grid variable (obtained from the PC analysis [20]), N_j is the number of logic cells in the j-th grid, $Slack_{j,k}$ is the slack of the k-th cell in the j-th grid, D_{nom} is the nominal critical delay of the circuit, and $\alpha < 0$ is a constant factor.

As a result, if a grid has many close-to-zero slack cells and/or its neighboring grids have many close-to-zero slack cells, the corresponding PC of that grid has a high criticality.

The PCs are then ordered, based on their criticalities and then assigned to the Sobol dimensions, sequentially. If there are more Sobol dimensions than PCs, the remaining Sobol dimensions are assigned to some of the non-correlated process parameters, according to a simple criticality measure for them, equal to $-1 \times slack_{cell}$. Thus, the smaller the slack of a cell is, the higher the probability that the non-correlated parameters of that cell are assigned to the Sobol samples.

VI. Results and Discussions

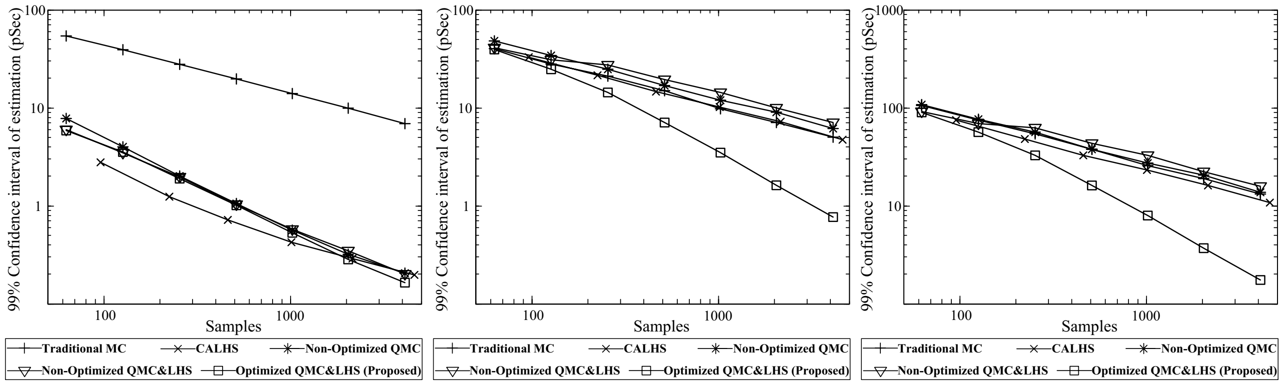
To verify the efficiency of the proposed technique, the ICCAS85 benchmark circuits are employed. The gate length variation is assumed to be Gaussian and spatially-correlated



(a) Dimensions $1, \cdots, 256$ (b) Dimensions $1, \cdots, 128$ (c) Dimensions $1, \cdots, 64$ (d) Dimensions $1, \cdots, 32$

(e) Dimensions $1, \cdots, 256$ (f) Dimensions $1, \cdots, 128$ (g) Dimensions $1, \cdots, 64$ (h) Dimensions $1, \cdots, 32$

Fig. 3. Distribution of L_2-discrepancies for 511 Sobol samples using (top) random initial values and (bottom) optimized initial values.



(a) Mean critical delay (b) Standard deviation of critical delay (c) 99th quantile point of critical delay

Fig. 4. Comparison of the confidence interval range of C6288 SSTA statistics by using different MC-based techniques.

with $\sigma_L = 12\%$ (the technique can generally accept any other type of distribution). In addition, the RDF-driven v_{th} variation is picked as the list of non-correlated random parameters. The results of the C6288 circuit, the largest benchmark, are chosen to provide a comparison of the algorithms in a high-dimensional case. All the other benchmarks exhibit the same superiority for the newly developed technique. The Capo [28] placer is selected to place the logic cells in order to determine the correlation coefficients of the spatial parameters. The area of the die of C6288 is partitioned into $11 \times 11 = 121$ grids. The timing response surfaces of the logic cells are characterized quadratically to deliver a high quality approximation in terms of process parameters based on a 65 nm CMOS industrial technology. The output rise/fall and the the gate propagation delay are expressed as functions of input rise/fall time, output load, gate length, and thereshold voltage.

The traditional MC, CALHS [3], traditional (non-optimized IV) QMC techniques, and mixed traditional QMC with the LHS techniques are compared with the proposed technique, a mixed low discrepancy (optimized IV) QMC with the LHS. The 99% confidence interval range of the mean, standard deviation, and 99-th quantile point of the C6288 critical delay are compared as the measures of the precision. This confidence interval is achieved by rerunning the experiments 2000 times for each technique and finding the standard deviation of the three reported statistics (mean, std, and 99-th quantile point). For the QMC-based samples, rerunning the original Sobol generator does not generate different sequences. Therefore, the scrambing technique, proposed for such purposes in [29], is used.

It is evident in Fig. 4 that except for the traditional MC, all the techniques perform well in the estimation of the mean of critical delay. However, only the proposed technique exhibits a significant superiority over the others including CALHS in estimating the critical delay variance and quantile point. Therefore, the 99th quantile point of the circuit's critical delay, the parameter of interest in the SSTA analysis, can be estimated more precisely by using the novel technique. It also can be seen that the convergence rate of the proposed technique, in estimating both the mean and standard deviation, approaches $O(N^{-1})$, when the number of samples increases, because the initial values are

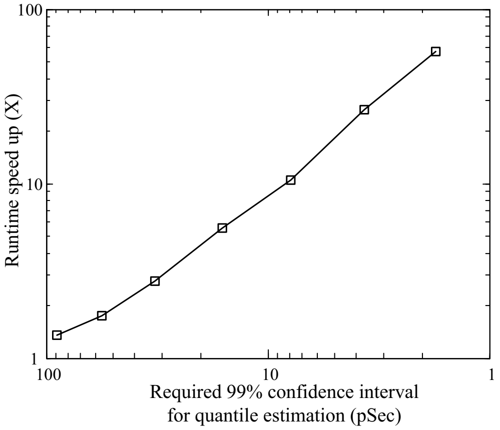


Fig. 5. The ratio of the runtime speedup of the proposed technique over the traditional MC to achieve a required confidence interval of the 99-th quantile point.

appropriately optimized to deal with an effectively 2-D problem.

Figure 5 shows the runtime speed up ratio of the proposed technique over the traditional MC versus the required confidence interval range of the quantile estimation. The ratio begins with 1.4X for a very wide confidence interval and increases almost linearly with the same rate of the required confidence range. The reason for such an aggressive (almost linear) increase in the runtime gain is that the ratio between the error of the MC and proposed technique is close to $O\left(N^{1/2}\right) = O\left(N^{-1/2}\right)/O\left(N^{-1}\right)$. Therefore, the number of samples to achieve a similar error as the traditional MC increases linearly, $O\left(N\right) = O\left((N^{1/2})^2\right)$.

Finally, it is noteworthy that, thanks to the unbiasness of LHS and QMC estimations, not only is the confidence interval range reduced in the proposed technique, but also are the actual values of the estimations (mean, standard deviation, and quantile point) unbiased and agree with the MC results.

VII. Conclusion

In this paper, an efficient MC-based SSTA technique is proposed that requires a significantly lower number of samples than the MC technique to provide statistical estimations with the same confidence interval range. The technique is established in relation to the fact that the variance of the circuit critical delays is strongly due to the pairwise interaction among the PCs of process parameters. This fact is verified by using a Legendre polynomial-based decomposition method. As a result, it is shown that the LHS technique, by itself, is not a suitable candidate for SSTA since LHS does not provide highly uniform samples in 2-D projections. The QMC sequence, Sobol, is chosen as another alternative; however, the poor pairings of some projections lead to no improvement in the variance estimation. Consequently, an optimization technique is proposed which manipulates the initial values, used in the Sobol generator, to provide samples with a higher 2-D uniformity. An SSTA technique is finally formed by combining the pre-optimized Sobol sequences and LH samples in a critical-aware framework. The significant reduction in required runtime for quantile estimation promises a faster timing sign-off of digital VLSI circuits.

Acknowledgment

This work was partially supported by NSERC. The information disclosed in this paper is patent pending.

References

[1] D. Blaauw et. al., “Statistical timing analysis: from basic principles to state-of-the-art,” *IEEE Trans. Computer-Aided Design*, vol. 27, pp. 589–607, Apr. 2008.

[2] L. Scheffer, “The count of Monte Carlo,” *ACM/IEEE TAU*, 2004.

[3] V. Veetil et. al., “Criticality aware Latin Hypercube Sampling for efficient statistical timing analysis,” *ACM/IEEE TAU*, personal communication.

[4] H. Niederreier, *Random number generation and quasi-Monte Carlo methods*. CBMS-NSF Regional Conference Series in Applied Math. no.63, SIAM, 1992.

[5] S. M. Ross, *Simulation*. 4th edition, Academic Press, 2006.

[6] M. D. McKay et. al., “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, pp. 239–245, 1979.

[7] J. F. Traub and A. G. Werschulz, *Complexity and information*. Cambridge University Press, 1998.

[8] P. Bratley and B. L. Fox, “Algorithm 659: Implementing Sobol’s quasirandom sequence generator,” *ACM Transactions on Mathematical Software*, vol. 14, no. 1, pp. 88–100, 1988.

[9] J. H. Halton, “On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,” *Numer. Math.*, 2, pp. 84–90, 1960.

[10] B. L. Fox, “Algorithm 647: Implementation and relative efficiency of quasirandom sequence generators,” *ACM Transactions on Mathematical Software*, vol. 12, no. 4, pp. 362–376, 1986.

[11] H. Niederreiter and C. Xing, “Low-discrepancy sequences and global function fields with many rational places,” *Finite Fields and Their Applications*, vol. 2, pp. 241–273, 1996.

[12] T. T. Warnock, “Computational investigations of low-discrepancy point sets II,” in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, Springer, pp. 354–361, 1995.

[13] I. H. Sloan and H. Wozniakowski, “When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?,” *Journal of Complexity*, vol. 14, pp. 1–33, 1998.

[14] A. Papageorgiou, “Sufficient conditions for fast quasi-Monte Carlo convergence,” *Journal of Complexity*, vol. 19, pp. 332–351, 2003.

[15] R. Caflisch et. al., “Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension,” *J. Comp. Finance*, vol. 1, pp. 27–46, 1997.

[16] X. Wang and I. H. Sloan, “Why are high-dimensional finance problems often of low effective dimension?,” *SIAM Journal on Scientific Computing*, vol. 27, no. 1, pp. 159–183, 2005.

[17] X. Wang and K. T. Fang, “The effective dimension and quasi-Monte Carlo integration,” *Journal of Complexity*, vol. 19, pp. 101–124, 2003.

[18] C. LeMieux and A. Owen, “Quasi-regression and the relative importance of the ANOVA components of a function,” in *Monte Carlo and Quasi-Monte Carlo Methods*, Springer, pp. 331–344, 2002.

[19] A. Jian and A. Owen, “Quasi-regression,” *Journal of Complexity*, vol. 17, pp. 588–607, 2001.

[20] H. Chang and S. Sapatnekar, “Statistical timing analysis under spatial correlations,” *IEEE Trans. Computer-Aided Design*, vol. 24, pp. 1467–1482, Sept. 2005.

[21] W. J. Morokoff and R. E. Caflisch, “Quasi-random sequences and their discrepancies,” *SIAM Journal on Scientific Computing*, vol. 15, pp. 1251–1279, 1994.

[22] D. E. Knuth, *The art of computer programming, vol. 2: Seminumerical Algorithms*. second edition, Addison-Wesley, 1981.

[23] P. Jaeckel, *Monte Carlo methods in finance*. Wiley, 2002.

[24] I. M. Sobol, “Uniformly distributed sequences with an additional uniform property,” *U.S.S.R. Computational Math. and Math. Phys. 16*, pp. 236–242, 1976.

[25] S. Joe and F. Y. Kuo, “Remark on algorithm 659: Implementing Sobol’s quasirandom sequence generator,” *ACM Transactions on Mathematical Software*, vol. 29, pp. 49–57, 2003.

[26] J. Cheng and M. J. Druzdzel, “Computational investigation of low-discrepancy sequences in simulation algorithms for Bayesian networks,” in Proc. *16th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 7281, 2000.

[27] S. Heinrich, “Efficient algorithms for computing the L2-discrepancy,” *Mathematics of Computation*, vol. 65, pp. 1621-1633, Oct. 1996.

[28] “Capo: A large-scale fixed-die placer from UCLA,” Available at: http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement.

[29] H. S. Hong and F. J. Hickernell, “Algorithm 823: Implementing scrambled digital sequences,” *ACM Transactions on Mathematical Software*, vol. 29, no. 2, pp. 95–109, 2003.