Bayesian Methods for the Yield Optimization of Analog and SRAM Circuits

Shuhan Zhang[1], Fan Yang[1*], Dian Zhou[2] and Xuan Zeng[1]
[1]State Key Lab of ASIC & System, School of Microelectronics, Fudan University, China
[2]Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, U.S.A

***Abstract*—As the technology node shrinks to the nanometer scale, process variation become one of the most important issues in IC designs. The industry calls for designs with high yield under process variations. Yield optimization is computationally intensive because traditionally it relies on the Monte-Carlo yield estimation. In this paper, we will first review the Bayesian methods that reduce the computational cost of yield estimation and optimization. By applying Bayes' theorem, maximizing the circuit yield is transformed to identify the design parameters with maximal probability density, conditioning on the event that the corresponding circuit is "pass". It can thus avoid repetitive yield estimations during optimization. The computational cost can also be reduced by using the Bayesian optimization strategy. By using the Gaussian process surrogate model and adaptive yield estimation, Bayesian optimization can significantly reduce the number of simulations while achieving even comparable yields for analog and SRAM circuits. We further propose a Bayesian optimization approach for yield optimization via max-value entropy search in this paper. The proposed max-value entropy search can better explore the state space, and thus reduce the number of circuit simulations while achieving competitive results.**

I. Introduction

As the IC technology node shrinks to the nanometer scale, the increasing process variations lead to server yield issues. For analog designers, the yield concern is especially emphasized, since the analog circuits are more sensitive to process variations. SRAM is a special kind of circuits with high yield requirements. An SRAM array contains millions of SRAM cells, the failure rate of an SRAM cell should be lower than 10^{-6} to guarantee the yield of the array. Efficient yield optimization approach is required to help the designers to optimize the yield during the design phase.

For yield optimization, repetitive yield estimations are required to guide the yield optimization. For yield estimation, a large number of simulations are generally needed to guarantee the accuracy, which would be very time-consuming. For SRAM circuits, to estimate the failure rate lower than 10^{-6}, the required number of simulations would be extremely large. Thus, we can improve the efficiency of yield optimization from two aspects. On the one hand, we could reduce the computational cost of each yield estimation. On the other hand, we could accelerate the optimization procedure.

Monte Carlo analysis is a general approach for yield estimation. A large number of circuit simulations are invoked to guarantee the accuracy of yield estimation for Monte Carlo analysis. A variety of approaches have been proposed to reduce the computational cost of Monte Carlo analysis. LHS [1] and Quasi-MC [2] methods employ the techniques of design of experiments to reduce the number of samples for Monte Carlo analysis. Prior knowledge from early design stages can also be utilized to reduce the required number of simulations in late stages for yield estimation. Bayesian inference has been proposed to fuse the prior knowledge from early stage and very few late-stage simulation samples to accurately estimate the yield of late stage in [3] [4]. Note that the accuracy of yield estimation could be low if the current yield is low. Thus, an optimal computation budget allocation (OCBA) technique is proposed in [5] [6]. Although this method can reduce the number of circuit simulations, the total computational cost of yield optimization is still high. For the yield estimation of SRAM circuits, many approaches have been proposed to reduce the required number of samples, e.g., important-sampling based approaches [7]–[10], boundary-based approaches [11]–[14], subset simulation [15] and scaled-sigma sampling [16].

To accelerate the optimization procedure, corner-based approaches have been proposed in [17]–[19]. The corner-based approaches try to optimize the worst-case performances over several corners. The computationally intensive yield estimation can thus be avoided. However, corner-based optimization would lead to over-design. Furthermore, the number of corners would also be extremely large if more process variations were considered. An alternative approach to accelerate the optimization process is building models to replace the circuit simulations [20] [21] [22] [23]. However, to guarantee adequate modeling accuracy, the required number of training samples would grow exponentially as the number of process variations increases.

Bayesian method is a powerful approach in machine learning community. It can encode the prior knowledge to the modeling process. The required number of training samples could thus be greatly reduced. The Bayes' theorem can be used to reformulate the problem, which would be more convenient to solve. We have proposed two Bayesian methods for yield optimization [24] [25]. In [24], we reformulate the yield optimization problem through Bayes' theorem. Instead of optimizing the yield, we aim to find the design parameters that maximize the probability density conditioning on the event that the corresponding circuit is "pass". The Gaussian kernel is used to approximate the probability density function and an EM-like algorithm is proposed to maximize the approximated probability density function. Through the reformulation, the repetitive yield estimations could be avoided. Nevertheless, the same computation efforts are allocated for both low-yield and high-yield candidates. It would lead to a waste of computational resources for low-yield candidates.

In [25], we proposed a yield optimization method based on Bayesian optimization and adaptive yield estimation. The Bayesian optimization [26] consists of two key ingredients, i.e., a probabilistic Gaussian process regression model and an acquisition function. The Gaussian process regression model assumes that the objective function could be modeled by a Gaussian process. The hyper-parameters of the Gaussian process could be obtained by the known training data. The predictions of the Gaussian process regression model include the expected yield but also the uncertainties of the prediction. This implies that our yield estimation is not necessarily to be accurate. The acquisition function is used to guide the optimization procedure. The expected improvement (EI) is taken as the acquisition function in this approach. It can be viewed as the expectation of the yield improvement based on the Gaussian process regression model. The acquisition function is designed to balance the exploitation and exploration.

*Corresponding author: yangfan@fudan.edu.cn.

978-1-7281-4123-7/20/$31.00 ©2020 IEEE

The exploitation means that the regions where the yields are expected to be high will be visited. The exploration means that the regions where the uncertainties of yields are high will be visited to guarantee the potential good solutions are not missed.

Bayesian optimization is a promising approach for yield optimization. On one hand, due to its efficient optimization strategy, it could significantly reduce the number of yield estimations. On the other hand, the Gaussian process regression model is possible to encode the observation uncertainties of yields, which enable us to adaptively control the yield estimation accuracy. Such an approach could significantly reduce the computational cost of yield estimation without sacrificing the accuracy of the final optimization results, since the uncertainties of the yield estimations are encoded in the Gaussian process regression model.

In this paper, we further propose a Bayesian optimization approach via max-value entropy search (MES) for yield optimization. The max-value entropy is proposed to replace the expected improvement (EI) acquisition function in [25]. By maximizing the max-value entropy search function, the search space can be better explored by maximizing the gain in mutual information between the optimum and the next query point. Compared with the traditional improvement-based acquisition functions, the experimental results demonstrate that the performance of MES function is competitive compared with state-of-the-art acquisition functions.

The rest of the paper is organized as follows. In section II, we will present the formulation of the yield optimization problem and review the existing Bayesian methods for yield optimization [24] [25]. Our proposed Bayesian optimization method via max-value entropy search is presented in section III. The efficiency of our proposed method is demonstrated in section IV. We conclude the paper in section V.

II. Background Review

In this section, we will review the problem formulation of the yield optimization problem firstly. Then, we will review two Bayesian methods for yield optimization [24] [25].

A. Problem Formulation

Denote $\mathbf{x} = [x_1, x_2, \cdots, x_{n_x}]^{\mathrm{T}} \in \chi$ an n_x-dimensional vector, which represents the design variables of the designs. χ is the design space which is specified by users. Denote $\mathbf{v} = [v_1, v_2, \cdots, v_{n_v}]^{\mathrm{T}} \in V$ an n_v-dimensional vector, which represents the n_v-dimensional process variations. V is the variation space. Generally, the v's are mutually independent and standard normal in most process design kits (PDK) provided by foundry.

Thus, the probability density function (PDF) of $\mathbf{v}$ can be expressed as

$$p(\mathbf{v}) = \prod_{i=1}^{n_v} \left[\frac{1}{\sqrt{2\pi}} \cdot \exp\Big(-\frac{1}{2} v_i^2 \Big) \right] = \frac{\exp(-\|\mathbf{v}\|_2^2/2)}{\big(\sqrt{2\pi}\big)^{n_v}}, \quad (1)$$

where $\|\cdot\|_2$ denotes the L_2-norm of a vector [27].

We consider a circuit performance c_i. It can be viewed as a function of $\mathbf{x}$ and $\mathbf{v}$. Without loss of generality, its specification could be written as $c_i \leq 0$. If K performances are considered, the circuit is expected to be "pass" if the K specifications are met. Denote the "pass" event as S, the conditional yield is expressed as

$$P(S|\mathbf{x}) = \int_V I(\mathbf{x}, \mathbf{v}) \cdot p(\mathbf{v}) \cdot d\mathbf{v}, \quad (2)$$

where $I(\mathbf{x}, \mathbf{v})$ is an indicator function. It equals to 1 if the circuit pass with design vector $\mathbf{x}$ and process vector $\mathbf{v}$, and 0 otherwise.

With these definitions, the yield optimization can be expressed as

$$\begin{array}{ll} \mathbf{x}_* = & \arg\max_{\mathbf{x}} P(S|\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \chi, \end{array} \quad (3)$$

where $\mathbf{x}_*$ is the optimal solution.

B. Reformulation of Yield Optimization via Bayes' Theorem

In [24], we proposed to reformulate the yield optimization problem via Bayes' Theorem. For design variables, we can view them as random variables uniformly distributed over the design space χ. Hence, $p(\mathbf{x})$ can be viewed as a constant over the design space χ.

By using Bayes' theorem, the conditional yield $\mathbf{x}$ can be rewritten as

$$P(S|\mathbf{x}) = \frac{P(S)}{p(\mathbf{x})} \cdot p(\mathbf{x}|S), \quad (4)$$

where $P(S)$ is expressed as

$$P(S) = \int_D \int_V I(\mathbf{x}, \mathbf{v}) \cdot p(\mathbf{v}) \cdot p(\mathbf{x}) \cdot d\mathbf{v} d\mathbf{x}, \quad (5)$$

and $p(\mathbf{x}|S)$ is the conditional probability density function (PDF) of $\mathbf{x}$

$$p(\mathbf{x}|S) = \frac{1}{P(S)} \cdot \int_V I(\mathbf{x}, \mathbf{v}) \cdot p(\mathbf{v}) \cdot p(\mathbf{x}) \cdot d\mathbf{v}. \quad (6)$$

Equation (4) is slightly different from the standard Bayes' theorem. In (4), the probabilities $P(\mathbf{x})$ and $P(\mathbf{x}|S)$ are replaced by their PDFs $p(\mathbf{x})$ and $p(\mathbf{x}|S)$.

Note that $P(S)$ is a constant over the combined space of χ and V. Also, $p(\mathbf{x})$ is a constant. Therefore, we have

$$P(S|\mathbf{x}) \propto p(\mathbf{x}|S). \quad (7)$$

It means that we can transform the yield optimization problem to an equivalent problem, which can be expressed as

$$\begin{array}{ll} \mathbf{x}_* = & \arg\max_{\mathbf{x}} p(\mathbf{x}|S) \\ \text{s.t.} & \mathbf{x} \in \chi. \end{array} \quad (8)$$

From (8), we can find that maximizing the yield is equivalent to identify $\mathbf{x}$ with the highest probability density, conditioning on the occurrence of the circuit "pass" event. In other words, to improve the yield, it is equivalent to find a design vector such that the number of "passed" designs is maximized in the neighborhood of this vector under process variations.

Since $p(\mathbf{x}|S)$ is unknown, we approximate $p(\mathbf{x}|S)$ by kernel density estimation. Firstly, N samples $\boldsymbol{\theta}_i = [\mathbf{x}_i \ \mathbf{v}_i]^{\mathrm{T}}, i = 1, \cdots, N$ were generated in the combined design space and process variation space. With circuit simulations, we can find M out of N samples are "pass". Their design vectors can be expressed as $\boldsymbol{\mu}_i, i = 1, \cdots, M$, which follows the distribution $p(\mathbf{x}|S)$. The distribution can be approximated by kernel density estimation. $p(\mathbf{x}|S)$ is approximated by probability density function $\hat{p}(\mathbf{x}|S)$, which can be expressed as

$$\hat{p}(\mathbf{x}|S) = \frac{1}{M} \cdot \sum_{i=1}^{M} K_{\mathbf{H}}(\mathbf{x} - \boldsymbol{\mu}_i), \quad (9)$$

where $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} \cdot K(\mathbf{H}^{-1/2}\mathbf{x})$. $K(\cdot)$ is the kernel function. $\mathbf{H}$ is an $n_x \times n_x$ bandwidth matrix. The multivariate Gaussian kernel is used here.

$$K(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^{n_x}} \cdot \exp\Big(-\frac{1}{2} \cdot \mathbf{x}^{\mathrm{T}}\mathbf{x} \Big). \quad (10)$$

Note that $\hat{p}(\mathbf{x}|S)$ is a Gaussian mixture distribution. We can use an expectation-maximization (EM) like algorithm from the machine learning community [28] to maximize $\hat{p}(\mathbf{x}|S)$. Please refer to [24] for the details of the EM-like algorithm.

C. Yield Optimization via Bayesian Optimization

The Bayesian optimization approach is a sequential model-based framework proposed for the optimization of noisy and expensive black-box functions. This implies that our yield estimation is not necessarily to be accurate. In [25], we proposed an efficient yield optimization method based on Bayesian optimization.

The Bayesian optimization [26] consists of two key ingredients, i.e., a probabilistic surrogate model and an acquisition function. Gaussian Process Regression (GPR) model is a widely used surrogate model. Consider n training data points $\mathbf{X}_n = \{\mathbf{x}_i, i = 1, \cdots, n\}$, and denote $f_i = f(\mathbf{x}_i)$ and $\mathbf{y}_n = \{y_i, i = 1, \cdots, n\}$ as the real yields and the noisy yield estimations, respectively. In GPR model, $f(\mathbf{x})$ is assumed to follow Gaussian process

$$f(\mathbf{x}) \sim \mathcal{GP}\big(\mu_0(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big), \tag{11}$$

where μ_0: $D \mapsto \mathbb{R}$ denotes the mean function, and k: $D \times D \mapsto \mathbb{R}$ denotes the covariance function.

According to the Gaussian process assumption, $\mathbf{f} = [f_1, f_2, \cdots, f_n]^{\mathrm{T}}$ follows a joint Gaussian distribution. We assume the deviation of the yield estimator y and f_i could be modeled by a Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$. Thus, we have

$$\mathbf{f} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{12}$$

$$\mathbf{y} \mid \mathbf{f}, \sigma_\epsilon^2 \sim \mathcal{N}(\mathbf{f}, \sigma_\epsilon^2 \mathbf{I}), \tag{13}$$

where $\mathbf{m} = [\mu_0(\mathbf{x}_1), \cdots, \mu_0(\mathbf{x}_n)]^{\mathrm{T}}$, and $\mathbf{I}$ is the identity matrix. $\mathbf{K}$ is the kernel matrix given by

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}. \tag{14}$$

For a new point $\mathbf{x}'$, the prediction f' could be obtained according to the assumption that f' and $\mathbf{f}$ also follow a joint Gaussian distribution. The prediction of f' is a Gaussian distribution $\mathcal{N}(\mu(x'), \sigma^2(x'))$, where

$$\mu(\mathbf{x}') = \mu_0(\mathbf{x}') + \mathbf{k}^{\mathrm{T}}\big(\mathbf{K} + \sigma_\epsilon^2 \mathbf{I}\big)^{-1}(\mathbf{y} - \mathbf{m})$$
$$\sigma^2(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \mathbf{k}^{\mathrm{T}}\big(\mathbf{K} + \sigma_\epsilon^2 \mathbf{I}\big)^{-1}\mathbf{k}.$$

The hyperparameters of the Gaussian process could be obtained by maximizing the likelihood of the training data. The next query point is selected by maximizing the expected improvement:

$$\mathbf{x}_* = \arg\max_{\mathbf{x}} \mathbb{E}(\mathbf{I}(\mathbf{x})), \tag{15}$$

where

$$\mathbf{I}(\mathbf{x}) = \begin{cases} f_{t+1}(\mathbf{x}) - y(\mathbf{x}^+) & f_{t+1}(\mathbf{x}) > y(\mathbf{x}^+) \\ 0 & \text{otherwise}, \end{cases} \tag{16}$$

where $f_{t+1}(\cdot)$ is the GP model of the yield after t iterations, $y(\mathbf{x}^+)$ the best observed yield.

With the expected improvement acquisition function and the probabilistic Gaussian process model, the optimization procedure could efficiently balance exploration and exploitation, and thus leads to promising designs with a very small number of yield estimations.

The errors of the yield estimations are included in the Gaussian process model as shown in (13). Therefore, although the yield estimation is not accurate, the potential optimal solutions will not be missed.

For Monte Carlo based yield estimation, the relationship between the sample size N and the accuracy of the estimation P_{MC} can be expressed as [29]

$$N \approx \frac{P_{MC} \cdot (1 - P_{MC}) \cdot k_\gamma^2}{\Delta P_{MC}^2}, \tag{17}$$

where ΔP_{MC} indicates the confidence interval. k_γ reflects the confidence level. From (17), we can estimate the required number of samples if the accuracy is specified. Also, we can estimate the accuracy of yield estimation for a specific number of samples. Equation (17) enables an adaptive scheme to dynamically control the accuracy of yield estimation during the optimization procedure, which could significantly reduce the number of circuit simulations. Please refer to [25] for the details of this method.

III. Bayesian Optimization via Max-value Entropy Search

In this section, we will present our proposed Bayesian optimization approach for yield optimization via max-value entropy search.

A. Max-value Entropy Search

Entropy search methods employ the information theory based criteria to select the next query point during the Bayesian optimization. Denote $D_t = \{\mathbf{X}_t, \mathbf{y}_t\}$ the training data at the t-th optimization step. Denote $\mathbf{x}_*$ the location $\mathbf{x}_* = \operatorname{argmax}_{\mathbf{x} \in D} f(\mathbf{x})$ which corresponds to the global optimum of $f(\mathbf{x})$. Entropy search aims to find the next query point that achieves maximal information gain or uncertainty reduction of $\mathbf{x}_*$. The corresponding entropy reduction function can be expressed as

$$\begin{aligned} ES(\mathbf{x}) &= I(\{\mathbf{x}, y\}; \mathbf{x}_* | D_t) \\ &= H(p(\mathbf{x}_* | D_t)) - E[H(\mathbf{x}_* | D_t \cup \{\mathbf{x}, y\})], \end{aligned} \tag{18}$$

where $H(\cdot)$ denotes the differential entropy of a continuous probability. $E[H(\mathbf{x}_* | D_t \cup \{\mathbf{x}, y\})]$ represents the expectation of the entropy if a new data point $(\mathbf{x}, y)$ is observed. The expectation is over $p(y|D_t, \mathbf{x})$.

The predictive entropy search [30] employs the symmetric formulation of (18), which can be expressed as below

$$PES(\mathbf{x}) = H(p(y|D_t, \mathbf{x})) - E[H(y|D_t, \mathbf{x}, \mathbf{x}_*)], \tag{19}$$

where the expectation is over $P(\mathbf{x}_*|D_t)$. However, $P(\mathbf{x}_*|D_t)$ is not analytically intractable, and the approximation of $P(\mathbf{x}_*|D_t)$ is computation expensive [31].

In this paper, we follow the idea of the above information-based policies. Instead of measuring the information of $\mathbf{x}$, we try to use the information of the maximum value $y_* = f(\mathbf{x}_*)$ [31]. We try to maximize the gain in mutual information between the maximum value $y_* = f(\mathbf{x}_*)$ and the next query point [31], which could be approximated analytically. The corresponding max-value entropy search function is expressed as

$$\begin{aligned} MES(\mathbf{x}) &= I(\{\mathbf{x}, y\}; y_* | D_t) \\ &= H(p(y|D_t, \mathbf{x}))) - E[H(y|D_t, \mathbf{x}, y_*)] \end{aligned} \tag{20}$$

In equation (21), the first term $p(y|D_t, \mathbf{x})$ follows the Gaussian posterior distribution $N(\mu_t(\mathbf{x}), \sigma_t^2(\mathbf{x}))$, which is analytically tractable. The second term can be approximated by a Monte

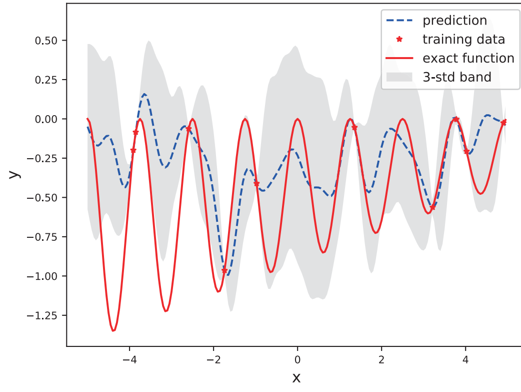

Fig. 1: The posterior distribution provided by the Gaussian process regression model with 1-hidden-layer neural network, and the number of neurons is set to be 50. The lightgray area represents the three standard deviation band.

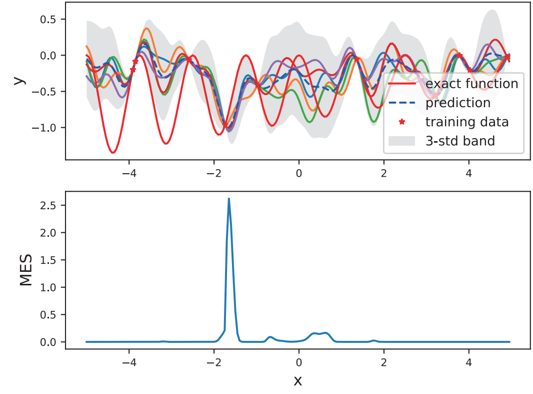

Fig. 2: The 5 sampled functions from the posterior distribution provided by the Gaussian process regression model with 1-hidden-layer neural network (top), and the corresponding max-value entropy search function (bottom).

Carlo estimation. Given the y_* of the underlying objective function, $p(y|D_t, \mathbf{x}, y_*)$ is a truncated Gaussian distribution. By sampling K functions out of the posterior distribution, we maximize each of them to generate a set of incumbent targets $Y_* = \{y_{*,1}, y_{*,2}, ..., y_{*,K}\}$. These incumbent targets can be used to approximate the second term of (21). The Monte Carlo estimation helps to generate a more accurate formulation for the mutual information gains between the underlying maximum value y_* and the candidate data points.

The $MES(\mathbf{x})$ can thus be approximated by [31]

$$MES(\mathbf{x}) \approx \frac{1}{K}\sum_{i=1}^{K}\frac{\lambda_i\psi(\lambda_i)}{2*\Psi(\lambda_i)} - log(\Psi(\lambda_i)), \tag{21}$$

where ψ and Ψ denote the probability density function and cumulative density function of normal distribution, respectively. Here, $\lambda_i = (\mu(\mathbf{x}) - y_{*,i}))/\sigma(\mathbf{x})$, and $y_{*,i}$ is the i-th sampled maximum value of the objective function.

B. Gaussian Process Regression Model with Neural Network

We employ a Gaussian process regression model with neural network to replace the original one. On the one hand, the computational cost of training and prediction of Gaussian process regression model can be reduced [23] with the neural network representation. On the other hand, the weight space view of neural network based Gaussian process model facilitates the Monte Carlo sampling of the predictions y_* in (21).

From a weight space view, the latent function $f(\mathbf{x})$ can be expressed as a linear combination of a nonlinear feature maps. Let $\phi(\cdot)$ represents the nonlinear function that maps data from D-dimensional input space to M-dimensional feature space. Denote y the observed data generated form the black-box function with additive noise with Gaussian distribution $\mathcal{N}(0, \sigma_n^2))$. $f(\mathbf{x})$ and y can be expressed as

$$\begin{cases} f(\mathbf{x}) = \boldsymbol{w}^T\phi(\mathbf{x}) \\ y \sim \mathcal{N}(f(\mathbf{x}), \sigma_n^2). \end{cases} \tag{22}$$

By characterizing the Gaussian prior of the weights $\boldsymbol{w}$ with zero mean and covariance matrix $\Sigma_p = \frac{\sigma_p^2}{M}I \in R^{M\times M}$, the distribution of weights follows $\boldsymbol{w} \sim \mathcal{N}(0, \Sigma_p)$ and the kernel function of the neural-network-based Gaussian Process regression model can be expressed as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\Sigma_p\phi(\mathbf{x}_j). \tag{23}$$

By assembling σ_n, σ_p and the weights of the neural network η into a single hyperparameter $\theta = (\sigma_n, \sigma_p, \eta)$, the model can be built by minimizing the reformulated negative log marginal likelihood function:

$$\begin{aligned} NLML = &\frac{1}{2\sigma_n^2}(\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\Phi^T A^{-1}\Phi\mathbf{y}) \\ &+ \frac{1}{2}log|A| + \frac{N}{2}log(2\pi\sigma_n^2) - \frac{M}{2}log\frac{M\sigma_n^2}{\sigma_p^2}. \end{aligned} \tag{24}$$

And the corresponding posterior distribution provided by the Gaussian Process regression model can be expressed as

$$\begin{cases} \mu(\mathbf{x}) = \phi(\mathbf{x})^T A^{-1}\Phi\mathbf{y} \\ \sigma^2(\mathbf{x}) = \sigma_n^2 + \sigma_n^2\phi(\mathbf{x})^T A^{-1}\phi(\mathbf{x}), \end{cases} \tag{25}$$

where

$$\begin{cases} \Phi = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_N)) \\ A = \Phi\Phi^T + \frac{M\sigma_n^2}{\sigma_p^2}I. \end{cases} \tag{26}$$

Figure 1 shows that the neural-network-based GPR model is able to capture the underlying behavior of the latent function with only a small set of training data.

From the above equations, the posterior distribution of the weights $\boldsymbol{w}$ follows the multivariate Gaussian $N(\mu(\boldsymbol{w}), \sigma^2(\boldsymbol{w}))$ which can be expressed as

$$\begin{cases} \mu(\boldsymbol{w}) = A^{-1}\Phi\mathbf{y} \\ \sigma^2(\boldsymbol{w}) = \frac{M\sigma_n^2}{\sigma_p^2}A^{-1}. \end{cases} \tag{27}$$

Fueled with equation (27), we can sample the i-th incumbent target $y_{*,i}$ by maximizing the i-th sampled function $\widetilde{f}(\mathbf{x}) = \boldsymbol{w}_i^T\phi(\mathbf{x})$. The sampled functions and the corresponding max-value entropy search function is shown in Figure 2. The MES is able to carefully balance between the exploitation and exploration to make the search efficient.

C. Summary

With the carefully designed MES function and neural-network-based Gaussian process regression model, the solution space of the latent function can be fully explored efficiently. By sampling a set of functions from the Gaussian posterior distribution and maximizing each of them, a set of incumbent targets can be generated to facilitate the Monte

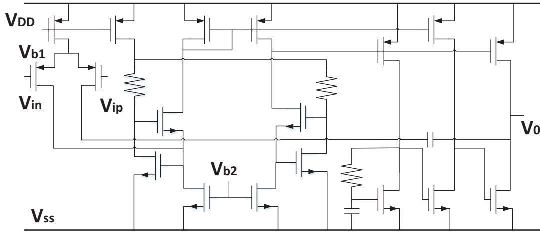

Fig. 3: The schematic of the three-stage amplifier, which is reproduced from [25].

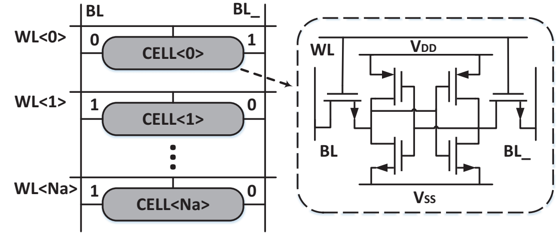

Fig. 4: The simplified schematic of an SRAM column, which is reproduced from [25].

TABLE I: Experimental results of the three-stage amplifier. The results of EI, BYO and ORDE are taken from [25].

	Algo	MES	EI	BYO	ORDE
Yield	best	**99.99%**	99.99%	99.99%	99.99%
	worst	**99.87%**	99.63%	97.57%	94.72%
	mean	**99.97%**	99.84%	99.17%	98.30%
	Std. Dev.	**0.04%**	0.13%	0.90%	2.18%
# Simulation	best	**3809**	3947	16400	37634
	worst	**12835**	24860	67100	163837
	mean	**8350**	10113	36260	81104

Carlo estimation. At each iteration, the query point is chosen by maximizing the mutual information gains between the maximum value of the latent function and the next query point. And the overall framework of the Bayesian optimization via max-value entropy search is presented in Algorithm 1.

Algorithm 1 Bayesian optimization via Max-value Entropy Search

1: Randomly sample a initial training set $D_0 = \{X, \mathbf{y}\}$
2: **for** t=1 to n **do**
3: Build the neural-network-based GPR model
4: Draw K functions from the Gaussian posterior distribution and maximize each of them to generate K incumbent targets $Y_* = \{y_{*,1}, y_{*,2}, ..., y_{*,K}\}$
5: Maximize the MES function and get $\mathbf{x}_t$
6: Estimate yield y_t at $\mathbf{x}_t$
7: Update the training dataset with the new observed data $\{\mathbf{x}_t, y_t\}$
8: **end for**

IV. Experimental Results

In this section, we conduct experiments on a three-stage amplifier and an SRAM circuit to demonstrate the efficiency of our proposed algorithm (denoted as MES) by comparing it to the following four state-of-the-art yield optimization algorithm: (1) EI [25] which employs the weighted expected improvement acquisition function to cut down the number of circuit simulations, (2) BYO [24] which tries to avoid repetitively invoking the yield estimations with reformation of the yield problem [32], (3) ORDE [5] which instead reduces the computational costs by combining the differential evolutionary algorithm with OCBA, (4) and AOSM [33], which accelerates the SRAM yield optimization procedure with adaptive online surrogate model. All experiments are conducted on a Linux workstation with 2 Intel Xeon CPUs and 128 GB memory.

A. Three-Stage Amplifier

The three-stage amplifier circuit is implemented in a 0.35 μm CMOS process, and the corresponding schematic is presented in Figure 3. A total of 24 design variables are considered in this circuit, and the design specifications are listed as follows

$$\begin{cases} GM > & 20dB \\ GBW > & 0.9MHz \\ PM > & 50^o \\ I_q < & 70\mu A. \end{cases} \tag{28}$$

In (28), GM denotes the gain margin, GBW represents gain-bandwidth, PM means phase margin and I_q is the quiescent current at 27°C. To ensure a fair comparison, we run each algorithm 10 times to reduce random fluctuations. And the final yields are estimated with 50000 Monte Carlo simulations.

The yield results and the number of circuit simulations are presented in Table I. The final optimization results achieved by our MES are competitive compared to the state-of-the-art, while establishing a much more stable performance. As for the number of circuit simulations, a 17.4% reduction is achieved compared to EI on average. And the speed up with respect to BYO and ORDE are 77.0% and 89.7%, respectively. It is also worth noting that MES significantly reduces the worst simulation times (e.g., by 48.4% compared to EI). This demonstrates its stability and efficiency.

B. SRAM Circuits

The simplified schematic of an SRAM column is presented in Figure 4, where N_a denotes the number of replicated SRAM bit cells implemented with a 45nm CMOS process. In this experiment, we set the I_{read} as the performance of interest. And a total of six design variables are considered, including the widths and lengths of the drive, access and load transistors. The experiments are conducted for $N_a = 1$, $N_a = 64$ and $N_a = 512$ to test both low-dimensional and high-dimensional variation space cases. Again, to ensure a fair and meaningful comparison, each algorithm is run 10 times to reduce random fluctuations. And the final failure rates are estimated with 10^8 Monte Carlo simulations.

The failure rates and the average simulation times are presented in Table II. For $N_a = 512$, the AOSM algorithm fails in the optimization process, due to the unaffordable modeling cost for high dimensionality. In both low- and high-dimensional variation space cases, MES achieves lower failure rates compared with the state-of-the-art algorithms. And the failure rates for $N_a = 512$ are reduced for 55.1% with respect to both EI and BYO algorithm. As for the number of circuit simulations, MES speeds up the optimization process by 33.0% compared to EI in low-dimensional variation spaces when $N_a = 1$. And the number of circuit simulations is reduced by 13.1% when $N_a = 64$. This again shows that MES outperforms the state-of-the-art algorithms in both stability and efficiency.

TABLE II: The optimization results of the SRAM. The results of EI, BYO and AOSM are taken from [25].

	Algo	MES	EI	BYO	AOSM
$N_a = 1$	Failure Rate	**2.96e-7**	3.43e-7	3.45e-7	3.45e-7
	# Simulation	**3722**	5556	8100	14750
$N_a = 64$	Failure Rate	**3.36e-7**	3.80e-7	3.81e-7	3.80e-7
	# Simulation	**5399**	6214	8100	640146
$N_a = 512$	Failure Rate	**3.51e-7**	7.82e-7	7.82e-7	-
	# Simulation	**6138**	6450	12600	-

V. Conclusion

In this paper, we review the state-of-the-art yield optimization algorithms and propose an efficient Bayesian optimization approach for yield optimization via max-value entropy search. The proposed algorithm is able to better search the state space with the max-value entropy search acquisition function, and reduce the number of circuit simulations while achieving competitive results. Experimental results demonstrate the stability and efficiency of our proposed method.

Acknowledgements

This research was supported by National Natural Science Foundation of China (NSFC) research projects 61822402, 61574044 and 61574046.

References

[1] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[2] Amith Singhee, Sonia Singhal, and Rob A Rutenbar. Practical, fast monte carlo statistical static timing analysis: why and how. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 190–195. IEEE Press, 2008.

[3] Xin Li, Wangyang Zhang, Fa Wang, Shupeng Sun, and Chenjie Gu. Efficient parametric yield estimation of analog/mixed-signal circuits via bayesian model fusion. In *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 627–634. IEEE, 2012.

[4] Chenlei Fang, Fan Yang, Xuan Zeng, and Xin Li. Bmf-bd: Bayesian model fusion on bernoulli distribution for efficient yield estimation of integrated circuits. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2014.

[5] Bo Liu, Francisco V Fernández, and Georges GE Gielen. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(6):793–805, 2011.

[6] Ivick Guerra-Gomez, Esteban Tlelo-Cuautle, and Luis Gerardo de la Fraga. Ocba in the yield optimization of analog integrated circuits by evolutionary algorithms. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1933–1936. IEEE, 2015.

[7] Rouwaida Kanj, Rajiv Joshi, and Sani Nassif. Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events. In *2006 43rd ACM/IEEE Design Automation Conference*, pages 69–72. IEEE, 2006.

[8] Lara Dolecek, Masood Qazi, Devavrat Shah, and Anantha Chandrakasan. Breaking the simulation barrier: SRAM evaluation through norm minimization. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 322–329. IEEE Press, 2008.

[9] Masood Qazi, Mehul Tikekar, Lara Dolecek, Devavrat Shah, and Anantha Chandrakasan. Loop flattening & spherical sampling: Highly efficient model reduction techniques for sram yield analysis. In *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, pages 801–806. IEEE, 2010.

[10] Shupeng Sun, Yamei Feng, Changdao Dong, and Xin Li. Efficient sram failure rate prediction via gibbs sampling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(12):1831–1844, 2012.

[11] Jian Yao, Zuochang Ye, and Yan Wang. Importance boundary sampling for sram yield analysis with multiple failure regions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(3):384–396, 2014.

[12] Fang Gong, Hao Yu, Yiyu Shi, Daesoo Kim, Junyan Ren, and Lei He. Quickyield: An efficient global-search based parametric yield estimation with performance constraints. In *Design Automation Conference*, pages 392–397. IEEE, 2010.

[13] Chenjie Gu and Jaijeet Roychowdhury. An efficient, fully nonlinear, variability-aware non-monte-carlo yield estimation procedure with applications to sram cells and ring oscillators. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 754–761. IEEE Computer Society Press, 2008.

[14] Shweta Srivastava and Jaijeet Roychowdhury. Rapid estimation of the probability of sram failure due to mos threshold variations. In *2007 IEEE Custom Integrated Circuits Conference*, pages 229–232. IEEE, 2007.

[15] Shupeng Sun and Xin Li. Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, pages 324–331. IEEE Press, 2014.

[16] Shupeng Sun, Xin Li, Hongzhou Liu, Kangsheng Luo, and Ben Gu. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(7):1096–1109, 2015.

[17] Robert Schwencker, Frank Schenkel, Michael Pronath, and H Graeb. Analog circuit sizing using adaptive worst-case parameter sets. In *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, pages 581–585. IEEE, 2002.

[18] Manidip Sengupta, Sharad Saxena, Lidia Daldoss, Glen Kramer, Sean Minehane, and Jianjun Cheng. Application-specific worst case corners using response surfaces and statistical models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(9):1372–1380, 2005.

[19] Manuel Barros, Jorge Guilherme, and Nuno Horta. Analog circuits optimization based on evolutionary computation techniques. *Integration*, 43(1):136–155, 2010.

[20] Shubhankar Basu, Balaji Kommineni, and Ranga Vemuri. Variation-aware macromodeling and synthesis of analog circuits using spline center and range method and dynamically reduced design space. In *2009 22nd International Conference on VLSI Design*, pages 433–438. IEEE, 2009.

[21] Jian Yao, Zuochang Ye, and Yan Wang. An efficient sram yield analysis and optimization method with adaptive online surrogate modeling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(7):1245–1253, 2014.

[22] Shuhan Zhang, Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, Xuan Zeng, and Xiangdong Hu. An efficient multi-fidelity bayesian optimization approach for analog circuit synthesis. In *Proceedings of the 56th Annual Design Automation Conference 2019*, page 64. ACM, 2019.

[23] Shuhan Zhang, Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Bayesian optimization approach for analog circuit synthesis using neural network. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1463–1468. IEEE, 2019.

[24] Mengshuo Wang, Fan Yang, Changhao Yan, Xuan Zeng, and Xiangdong Hu. Efficient bayesian yield optimization approach for analog and SRAM circuits. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17, pages 11:1–11:6, New York, NY, USA, 2017. ACM.

[25] Mengshuo Wang, Wenlong Lv, Fan Yang, Changhao Yan, Wei Cai, Dian Zhou, and Xuan Zeng. Efficient yield optimization for analog and sram circuits via gaussian process regression and adaptive yield estimation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(10):1929–1942, 2018.

[26] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[27] Shupeng Sun, Xin Li, Hongzhou Liu, Kangsheng Luo, and Ben Gu. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 478–485. IEEE, 2013.

[28] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[29] Helmut E Graeb. *Analog design centering and sizing*, volume 64. Springer, 2007.

[30] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

[31] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3627–3635. JMLR. org, 2017.

[32] SK Au. Reliability-based design sensitivity by efficient simulation. *Computers & structures*, 83(14):1048–1061, 2005.

[33] Jian Yao, Zuochang Ye, and Yan Wang. An efficient SRAM yield analysis and optimization method with adaptive online surrogate modeling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(7):1245–1253, 2015.