A Global Bayesian Optimization Algorithm and Its Application to Integrated System Design

Hakki Mert Torun, *Student Member, IEEE*, Madhavan Swaminathan, *Fellow, IEEE*, Anto Kavungal Davis, and Mohamed Lamine Faycal Bellaredj

***Abstract*— Increasing levels of system integration pose difficulties in meeting design specifications for high-performance systems. Oftentimes increased complexity, nonlinearity, and multiple tradeoffs need to be handled simultaneously during the design cycle. Since components in such systems are highly correlated with each other, codesign and co-optimization of the complete system are required. Machine learning (ML) provides opportunities for analyzing such systems with multiple control parameters, where techniques based on Bayesian optimization (BO) can be used to meet or exceed design specifications. In this paper, we propose a new BO-based global optimization algorithm titled Two-Stage BO (TSBO). TSBO can be applied to black box optimization problems where the computational time can be reduced through a reduction in the number of simulations required. Empirical analysis on a set of popular challenge functions with several local extrema and dimensions shows TSBO to have a faster convergence rate as compared with other optimization methods. In this paper, TSBO has been applied for clock skew minimization in 3-D integrated circuits and multiobjective co-optimization for maximizing efficiency in integrated voltage regulators. The results show that TSBO is between 2×-4× faster as compared with previously published BO algorithms and other non-ML-based techniques.**

***Index Terms*— 3-D integration, Bayesian optimization (BO), black box systems, integrated voltage regulator (IVR), machine learning (ML), magnetic core inductor, thermal management.**

I. Introduction

EMERGING trends in advanced electronics include high levels of system integration to maximize electrical performance. Popular techniques include 3-D integration to overcome electrical performance and density problems [1], [2] and integrated voltage regulators (IVRs) to address needs of high-performance processors that support dynamic voltage and frequency scaling for energy-efficient multicore architectures [3], [4].

As the scale of integration increases, it raises new challenges in design related to modeling and optimization. Usually, the tuning process for such systems to meet design specifications requires substantial effort due to high levels of nonlinearity and crosstalk between different components comprising the system. Global optimization methods are often used to address the tuning of such systems. However, many of the optimization algorithms are prone to challenges related to initial point selection, computation or approximation of gradients, surrogate model generation, and slow convergence rates.

Machine learning (ML), namely Bayesian optimization (BO), is a well-known method for optimizing expensive black box functions where a closed-form expression or surrogate model is unavailable [5], [6]. Prior application of BO to electronics design and tuning shows that it provides better convergence rates as compared with typical methods used in electronics design automation (EDA) [7], [8]. However, BO-based algorithms in prior work have been based on general purpose algorithms which do not specifically address the EDA-related challenges, such as increased number of control parameters and focusing on reducing number of simulations instead of algorithm progression time. Majority of the EDA problems, including 3-D ICs and IVRs, require CPU intensive simulations of multiscale and multiphysics structures that can consume significant CPU time. In this paper, we therefore propose a new BO-based global optimization algorithm, namely Two-Stage BO (TSBO), developed specifically with the purpose of reducing the computational overhead by reducing number of simulations required to find the optimal design point as desired for problems encountered in EDA. In [9], we showed very preliminary results of TSBO on temperature gradient minimization of 3-D ICs, and in [10], we included the clock skew minimization as our optimization goal. In this paper, we extend [9], [10] by providing extensive algorithmic details of TSBO and show the performance of the algorithm for two applications, namely: 1) thermal management for 3-D ICs with an ultimate goal of skew minimization [8] and 2) co-optimization for maximizing the efficiency of IVR.

The rest of this paper is structured as follows. Section II provides theoretical details on black box optimization and BO; Section III presents the TSBO algorithm with application to known functions; Section IV shows the application of the algorithm presented for clock skew minimization in 3-D ICs by minimizing temperature gradients; Section V addresses multiobjective optimization of IVR followed by the conclusion in Section VI.

II. Black Box Optimization

Nonconvex optimization attracts attention in many fields of engineering problems, since a major portion of realistic

Manuscript received July 17, 2017; revised November 10, 2017; accepted December 11, 2017. Date of publication January 11, 2018; date of current version March 20, 2018. This work was supported in part by the NSF I/UCRC Center for Advanced Electronics through Machine Learning (CAEML) and in part by the Power Delivery for Electronic Systems Consortium, Georgia Tech, USA. *(Corresponding author: Hakki Mert Torun.)*

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: htorun3@gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2017.2784783

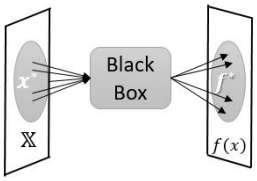


Fig. 1. Optimization of black box systems.

systems do not tend to have linear responses to its control parameters. The problem can be posed from the perspective of optimization as

$$\begin{aligned} &\max_{x\in\mathbb{X}^d} f(x) \quad \text{or} \quad \min_{x\in\mathbb{X}^d} f(x) \\ &\text{where } x \in \mathbb{X} \subset \mathbb{R} \end{aligned} \tag{1}$$

where x is the d dimensional input vector (control parameters) and $f(x)$ is the objective function.

In the domain of advanced electronics design, $f(x)$ can represent a high-dimensional, highly coupled, and multiscale system, such as 3-D ICs, IVRs with package-embedded passives, or high-speed channels. As it is not trivial to develop accurate and precise models of such systems, a necessity of approaching the problem in (1) using a black box setting arises as shown in Fig. 1.

A. Bayesian Optimization Based on Gaussian Process

In majority of black box systems, including 3-D ICs, it is not possible directly to access gradient information of $f(x)$ at an arbitrary x. Using BO based on Gaussian Processes (GPs), gradient information is not required; hence, it becomes a suitable and promising candidate for black box optimization. BO is a well-known method in the ML community and has been mainly used for hyperparameter tuning for various ML algorithms. It is as an active learning algorithm, as in the process of optimization, BO selects the next observation to maximize reward toward finding the solution of (1). The approach is based on Bayes' Theorem

$$P(f|D_{1:t}) \propto P(D_{1:t}|f)P(f) \tag{2}$$

where $D_{1:t} = [(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)]$ represent observations made so far; $P(f)$, $P(f|D_{1:t})$, and $P(D_{1:t}|f)$ are probabilities of prior, posterior, and likelihood, respectively.

Although other distributions and predictive models can be used as priors in BO, one of the most commonly used is GP due to its theoretical and practical advantages. For instance, it has been theoretically shown that Bayesian neural networks, which provides a posterior over network's weights, with one hidden layer converges to a GP as the number of hidden units go to the infinity [11]. Moreover, GPs with certain type of kernels have been shown to have universality property over a compact set, aligning with the objective of BO [12]. From practical point of view, being a nonparametric method, GPs eliminate the need for determining problem-specific model parameters encountered in neural networks, such as network architecture, number of hidden layers and hidden units, learning rate, and activation function, to be used. A more detailed description for advantages of GP over other methods for predictive modeling can be found in [13].

In BO, the underlying function to be optimized, $f(x)$, is represented as a joint, multidimensional GP with a mean (μ) and covariance (K) matrix, given by

$$f_{1:t} = \mathcal{N}(\mu(x_{1:t}), K(x_{1:t})). \tag{3}$$

This step of BO is where the training of the predictive GP occurs. It should also be noted that unlike other ML algorithms, the objective in BO is not to derive a predictor covering the entire sample space but only accurately predict where global extrema lies in the sample space using the previous observations.

As the problem in (1) is to be solved in a black box setting, it is assumed that there is no prior information about the underlying function. Hence, we choose a zero mean GP for $P(f)$ and use one of the popular kernel functions used in the literature, namely automatic relevance determination (ARD) Matern 5/2 function, given as

$$K(x) = \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \ldots & k(x_t, x_t) \end{bmatrix} \tag{4}$$

$$k(x_i, x_j) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) e^{-\sqrt{5}r} \tag{5}$$

where $r = (\sum_{d=1}^{D}((x_{i,d} - x_{j,d})^2/\sigma_d^2))^{1/2}$; σ_f and σ_d are hyperparameters of $K(x)$ which are updated during the training process to minimize the negative log marginal likelihood of the GP. In addition, using *ARD* type kernel has several advantages. As each parameter has its own scaling parameter to be determined in *ARD* type kernels, they enable interpretability of the underlying problem and act as an implicit sensitivity analysis. For instance, after the training of predictive GP is completed, if σ_d of one parameter is sufficiently larger compared with other parameters, it can be said that a change in this parameter has lower sensitivity on the prediction. Another possible use of σ_d parameters in (5) is transferring problem-specific knowledge during the training of GP. If, for example, a certain parameter is believed to have higher effect on the output, the training process can be modified to force σ_d of that parameter to be smaller compared with others. However, in high-dimensional problems encountered in EDA, this can be very dangerous as predicting the importance of each parameter before having any prior training data can result in suboptimal designs.

The traditional approach of BO is obtaining an acquisition function, $u(x)$, based on a predefined strategy and using auxiliary optimization on this acquisition function to find the new query point, x_{t+1}. Note that optimizing $u(x)$ does not require any additional queries, but uses the knowledge of previous samples to get a prediction at candidate points using

$$\mu(x_{t+1}) = k^T K^{-1} f_{1:t} \tag{6}$$

$$\sigma^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - k^T K^{-1} k \tag{7}$$

where K and k are given by (4) and (5), respectively. In the literature of BO, most prominent acquisition functions are

probability of improvement (PI), expected improvement (EI), and upper confidence bound (UCB) given as

$$u_{\mathrm{PI}} = \Phi((\mu(x) - \widetilde{f}^* - \zeta)/\sigma(x)) \quad (8)$$

$$u_{\mathrm{EI}} = (\mu(x) - \widetilde{f}^* - \zeta)\Phi(Z) + \sigma(x)\phi(Z) \quad (9)$$

$$u_{\mathrm{UCB}} = \mu(x) + K\sigma(x), \quad K = \sqrt{2ln(2\pi M^2/(12\eta))} \quad (10)$$

where $\widetilde{f}^*$ is the best point observed so far, ζ is a hyperparameter for u_{PI} and u_{EI}, M is number of calls made to UCB, $(1-\eta)$ is the probability of zero regret for GP-UCB [6], $Z = (\mu(x) - \widetilde{f}^* - \zeta)/\sigma(x)$, $\Phi(.)$ and $\phi(.)$ are the cumulative distribution function and probability density function of normal distribution, respectively.

A common situation encountered in the optimization of electronic systems is handling of design constraints. For instance, in the IVR example discussed, the range in the dimension of the physical parameters of the inductor could be constrained due to the process while area and efficiency constraints need to be satisfied in the final design. In the context of BO, there are several ways to introduce such design constraints in the optimization framework. One method is to penalize the objective function when the set of parameters lies in such unfeasible regions to prevent that point from being labeled as optimal. Another method is to modify the acquisition function so that the parameters in the unfeasible regions are less likely to be the next simulation point [14]. We refer readers to [15] for a more detailed theoretical background on BO and GPs.

As mentioned in Section I, prior work successfully applied BO to electronics design. Chen *et al.* [7] partially used BO with EI and PI criteria, to optimize broadband matching network of power amplifiers but used predetermined algorithms for generating vast amount of training data before starting optimization. Park *et al.* [8] used a complete BO approach to optimize temperature and temperature gradients of 3-D ICs, using the IMGPO algorithm [16]. While both works show significant improvement as compared with other non-ML-based methods, the algorithms used were not specifically developed for solving EDA problems with a focus on reducing number of simulations required.

III. Two-Stage Bayesian Optimization

In multiarmed bandit problems, the path to achieve optimal result goes through a tradeoff between exploration and exploitation. For the case of aforementioned acquisition functions, u_{PI} and u_{EI} in (8) and (9), the tradeoff is made with the introduction of the hyperparameter ζ. IMGPO handles the tradeoff with the assumption of existence of a tighter bound than UCB [16], and BamSOO takes the approach of eliminating regions where with high probability, the region does not contain global optima [17].

In the new algorithm, we propose TSBO and we approach this tradeoff in two stages, namely fast exploration stage and pure exploitation stage, which can also be interpreted as coarse and fine tuning, respectively. In order to address EDA-related challenges, we present two additional new techniques that increases convergence rate and extends the applicability of the algorithm to a variety of electronics design problems.

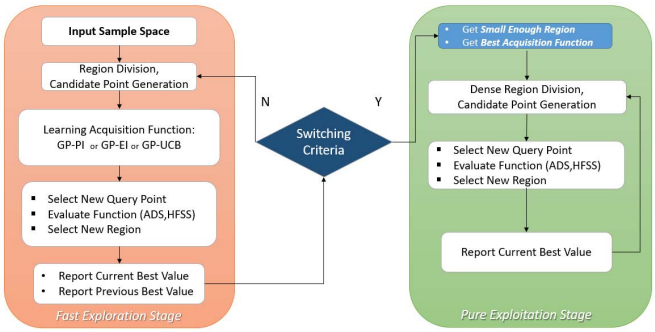


Fig. 2. Flowchart of the proposed algorithm.

These consist of: 1) sublearning process using acquisition functions and 2) a distinctive hierarchical partitioning tree construction scheme. The flowchart of TSBO is given in Fig. 2.

A. Fast Exploration and Pure Exploitation Stages

The first stage of TSBO, *fast exploration stage*, can be considered as coarse tuning, as the purpose of this stage is to rapidly find the region, $\mathbb{A}^d$, in the sample space where the global optimum, $x^* = \mathrm{argmax}_x f(x)$, is contained. For the purpose of finding a tight $\mathbb{A}^d$ rapidly, we divide the sample space $\mathbb{X}$ into 2^d regions of hyperrectangles, H_t, and generate candidate points, $c_{i,j}$, to determine the next sampling point, x_{t+1}. These candidate points are chosen as the center of each region, given as

$$c_{t,j} = \frac{H_{t,j,\min} + H_{t,j,\max}}{2}, \quad j = 1, 2, \ldots, t2^d \quad (11)$$

where $c_{t,j}$ is the jth candidate point of a total of $t2^d$ points at iteration t; $H_{t,j}$ is the region that $c_{t,j}$ belongs to and $H_{t,j,\min}$ and $H_{t,j,\max}$ are corresponding lower and upper boundaries of each d dimensional region. In order to avoid relying on auxiliary optimization to select x_{t+1}, $u(x)$ is not optimized but evaluated at each candidate point as follows:

$$c_{(t,j^*)} = \arg\max_{c \in C} u^{(i)}(c) \quad (12)$$

$$x_{t+1} = c_{(t,j^*)} \quad (13)$$

where j^* denotes the selected candidate; C represents the finite set of candidate points and $u^{(i)}(x)$ is u_{UCB}, u_{EI}, or u_{PI}, selected sequentially as explained later in this section on learning acquisition functions. After querying the function at x_{t+1}, the new set of regions are generated using

$$H_n = \bigcup_{i=1}^{2^d} h_i, \quad h_i \subset H_{(t,j^*)} \quad (14)$$

$$H_{t+1} = H_n \cup H_t \quad (15)$$

where H_n is the union of new regions, h_i, acquired by dividing H_{t,j^*} into 2^d hyperrectangles.

The first stage remains until it explores a small enough region ($x^* \in \mathbb{A}^d$) such that the Euclidean distance between previous best and current best input points are negligible.

This is illustrated in Fig. 2 as the switching criteria, given as

$$n = \begin{cases} n+1, & \text{if } \|x_{\max} - x_{p\max}\| < 10^{-3} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $x_{\max}$ and $x_{p\max}$ are the input vector of current and previous best observations, respectively, and switching occurs if $n > N$. The selection of $\mathbb{A}^d$ is illustrated in Fig. 5(c) and given as

$$\mathbb{A}^d = \begin{bmatrix} (1-\alpha)x_{\max,1} & (1+\alpha)x_{\max,1} \\ \vdots & \vdots \\ (1-\alpha)x_{\max,d} & (1+\alpha)x_{\max,d} \end{bmatrix} \tag{17}$$

where $x_{\max,i}$ is the ith dimension of $x_{\max}$ in (16) and α is a hyperparameter of TSBO for choosing the tightness of the region provided to the second stage.

The second stage of TSBO, namely pure exploitation stage, takes $\mathbb{A}^d$ and $u^*(x)$ from the first stage as inputs and performs fine tuning for the optimization problem, i.e., increases accuracy. At this stage, the tight region $\mathbb{A}^d$ is divided into three new regions along its longest dimension at each iteration and candidate points are generated at each region in the same fashion as the first stage, but using the learned acquisition function of $u^*(x)$.

B. Hierarchical Partitioning Scheme

TSBO uses a distinctive hierarchical partitioning scheme that differentiates it from general purpose BO algorithms and makes it more EDA-oriented. Typically, a hierarchical partitioning tree is constructed via fully committing to and expanding the selected branch, i.e., sampling each child node [18]. For example, BamSOO uses a region shrinking technique and the tree is expanded only in nondiscarded region [17]. However, the child node of the selected branch is fully expanded if it belongs to a nondiscarded region. In TSBO, the selected branch is expanded to generate 2^d candidate points, i.e., child nodes, but the sampling only occurs at the most promising child node; hence, only one function query occurs per iteration. This overcomes the limitation in the number of branches that can be generated as in BamSOO and IMGPO and allows for rapid coverage of the sample space. In other words, with the cost of a few seconds of algorithm run time due to storing $t2^d$ points at each iteration, TSBO substantially reduces the number of required simulations needed to find x^*. An example partitioning tree constructed using TSBO is shown in Fig. 3.

C. Learning Acquisition Functions

As the acquisition function is a predetermined strategy, there is no guarantee that a single acquisition function will outperform others for every problem. GP-Hedge is a BO algorithm that uses popular acquisition functions defined in (8)–(10) and uses a probability-based gain strategy to determine which function best fits the problem [19]. We adapt this idea proposed in [19], however, we alter it and call it learning acquisition functions. Instead of choosing acquisition functions from a distribution, TSBO observes how each function behaves using the same gain strategy, i.e., by collecting training data.

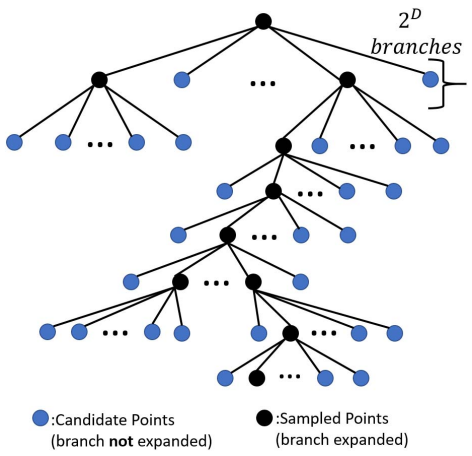


Fig. 3. Example partitioning tree constructed using TSBO.

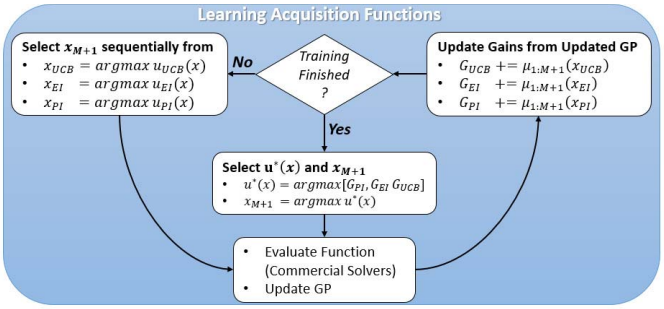


Fig. 4. Progression of learning acquisition function block at Fig. 2.

As summarized in Fig. 4, this sublearning process using acquisition functions uses PI, EI, and UCB criteria defined in (8)–(10) sequentially during training data collection. After M observations have been made, the algorithm quits the sequential strategy and starts to select $u^*(x)$ that has the largest gain so far. The gain functions for each $u(x)$ are continued to be updated at each iteration so that $u^*(x)$ is dynamically updated as simulations are performed.

Compared with GP-Hedge, BamSOO, and GP-UCB, this makes the algorithm deterministic along with the automatic initial point selection at the center of the sample space. The deterministic nature of TSBO makes it more robust.

D. Experiments on Challenge Functions

The progression of the proposed algorithm is provided in Fig. 5 and the objective is to maximize peaks function available in MATLAB. Here, candidate points cover all sample space, but sampling is concentrated on only promising regions. The region captured by the first stage [red rectangle in Fig. 5(c)] contains the global maximum, which is then used by the second stage to increase the accuracy up to 10^{-6}.

In order to test the performance of TSBO, we have considered four different challenge functions with different dimensions, which are common benchmarking problems in optimization [20]. Performance evaluation criteria in these experiments shown in Fig. 6 are log distance to optima, i.e., $\log_{10}(|f^* - \widetilde{f}^*|)$. Kawaguchi *et al.* [16] showed that

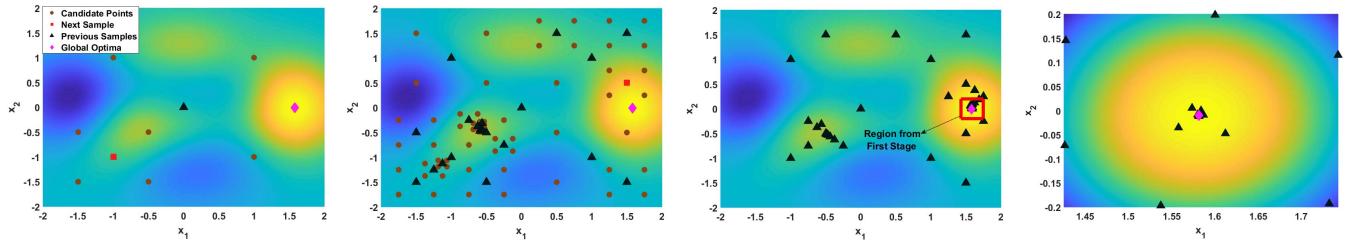


Fig. 5. Progression of TSBO for finding global optima of 2-D peaks function. (a) Starting point $t = 1$. (b) $t = 20$. (c) End of first stage $t = 42$. (d) End of second stage $t = 100$.

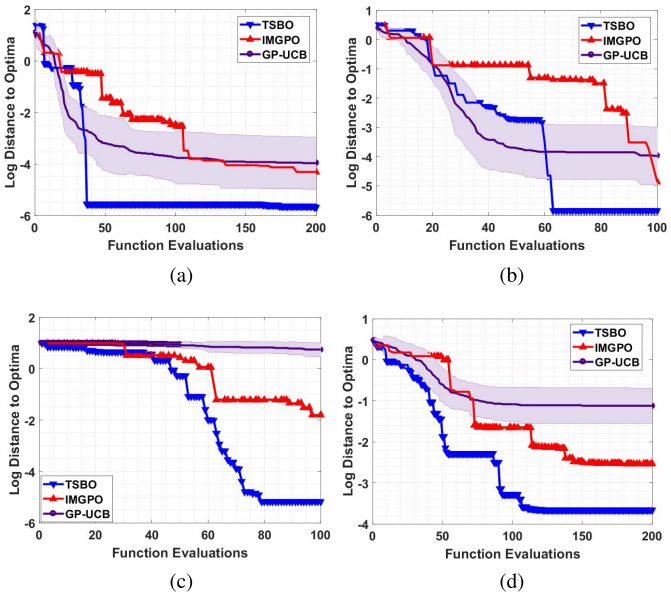


Fig. 6. Performance comparison for proposed algorithm, TSBO, on optimization challenge functions used in the literature. Functions are available in [20]. (a) 2-D Branin function. (b) 3-D Hartmann function. (c) 4-D Shekel5 function. (d) 6-D Hartmann function.

IMGPO outperforms BamSOO, GP-EI, and GP-PI on the same challenge functions used in this paper; hence, we compare TSBO with IMGPO and GP-UCB using MATLAB's *patternsearch* for auxiliary optimization of GP-UCB. Since the initial point is selected randomly in GP-UCB, we repeat the experiments 50 times and report the mean and standard deviation. As shown in Fig. 6, TSBO outperforms IMGPO and GP-UCB in these benchmark problems in terms of requiring less function evaluations for converging to the global optimum.

Furthermore, algorithm progression times (excluding function evaluation time) and total variable memory stored for 100 iterations are measured and compared in Table I. Although TSBO achieves higher accuracy with less function evaluations, it requires additional run time and memory to store and process the aforementioned $t2^d$ candidate points at each iteration. As IMGPO uses a partitioning scheme that has lighter weight on memory compared with TSBO, it requires lesser processing at each iteration which results in faster run times. GP-UCB uses the least amount of memory among the algorithms considered as it does not store additional variables at each iteration; however, its run time is the highest due to the auxiliary optimization process.

TABLE I
ALGORITHM PROGRESSION TIMES AND ACCURACY WITHIN 100 ITERATIONS

	GP-UCB		IMGPO		TSBO	
	Run Time (Var. Memory)	Accuracy	Run Time (Var. Memory)	Accuracy	Run Time (Var. Memory)	Accuracy
2D Branin	10.38s (0.99 Mb)	$10^{-3.8}$	1.28s (1.65 Mb)	$10^{-2.5}$	6.15s (2.85 Mb)	$10^{-5.6}$
3D Hartmann	10.52s (0.10 Mb)	10^{-4}	1.35s (1.66 Mb)	$10^{-4.8}$	4.35s (2.01 Mb)	$10^{-5.8}$
4D Shekel5	11.35s (0.10 Mb)	$10^{0.8}$	1.83s (1.68 Mb)	$10^{-1.8}$	6.79s (3.17 Mb)	$10^{-5.2}$
6D Hartmann	12.23s (0.11 Mb)	$10^{-1.2}$	1.91s (1.70 Mb)	$10^{-2.5}$	6.51s (2.19 Mb)	$10^{-3.3}$

TABLE II
RUN-TIME STATISTICS OF LEARNING ACQUISITION FUNCTIONS OF TSBO FOR CHALLENGE FUNCTIONS

	PI		EI		UCB	
	Normalized Gain	Number of Calls	Normalized Gain	Number of Calls	Normalized Gain	Number of Calls
2D Branin	0.2669	17	0.3164	16	0.4167	67
3D Hartmann	0.3205	16	0.3475	68	0.3321	16
4D Shekel5	0.4074	22	0.4228	62	0.1698	16
6D Hartmann	0.3352	16	0.3461	68	0.3207	16

In addition, run time statistics summarizing the effect of learning acquisition functions block of TSBO are provided in Table II. Here, the number of calls refers to number of times that particular $u(x)$ has been used to select x_{t+1}. It can be seen that each $u(x)$ contributes to the optimization process to find x^*; however, the amount of information gained from each $u(x)$ is different for each problem. As a result, the acquisition function providing the highest gain is called more by TSBO. Although the gains for each $u(x)$ are being dynamically updated after M observations as explained in Section III-C, the choice of $u^*(x)$ at M^{th} observation is only updated for the case of 4D-Shekel5 function as more observations are added.

The choice of hyperparameters can significantly affect the performance of black-box optimization algorithms, including TSBO. Here, problem-specific knowledge can be leveraged

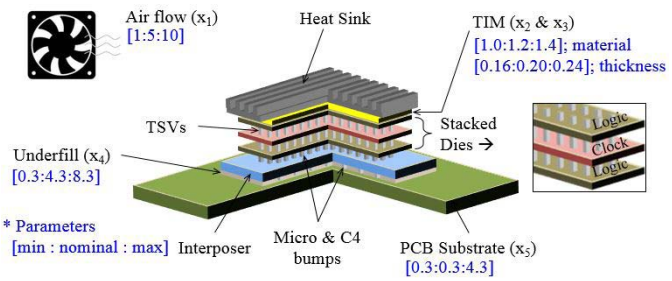


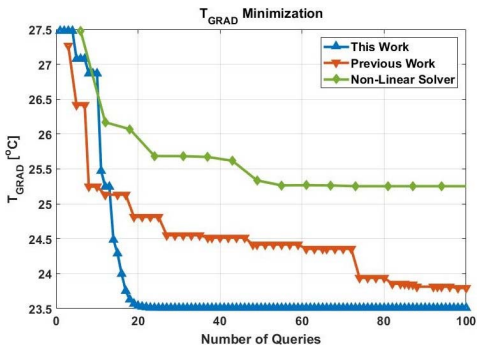
Fig. 7. Geometry of 3-D IC used for optimization [8].

to adjust these parameters. For instance, if the problem is believed to have large number of local extrema, the algorithm can be adjusted to give more importance to exploration than exploitation. For the case of BO-based algorithms, the effect of hyperparameters is a well-studied subject, resulting in rule of thumb values [15], [21], [22]. Counterintuitively, these studies suggest that the dynamic modulation of these parameters has little to no-effect on empirical performance of the algorithms. Following these works, for these experiments, the hyperparameters of TSBO are chosen as: $\alpha = 0.1$ in (17); $\eta = 0.1$ in (10); $\zeta = 0$ and $\zeta = 0.01$ in (8) and (9) respectively; $M = 50$ for learning acquisition functions and $N = 10$ for switching criteria. TSBO specific parameters, such as N, M, and α can be modified accordingly to transfer domain expertise to the algorithm. For instance, if the design under consideration is to be fabricated using a process that cannot provide accuracy beyond micrometer, fine tuning beyond this level using the second stage of TSBO may result in invalid designs. Hence, N can be increased to force TSBO to spend more time in the first stage.

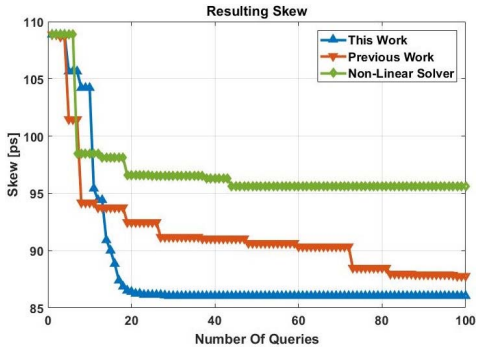
IV. Application I: 3-D Integrated Circuits

The first application chosen to evaluate the proposed algorithm is thermal management of 3-D ICs, as shown in Fig. 7. 3-D integration results in increased temperature and temperature gradient that degrades signal integrity. Furthermore, the multiscale geometry of the 3-D ICs requires CPU extensive simulations as finer mesh grids are used to capture variations in stacked dies connected with through silicon vias. Previous work has presented and verified an accurate modeling and simulation platform that uses an electrical–thermal solver [23] to calculate thermal characteristics and used BO to control temperature and temperature gradients to minimize clock skew [8]. In this paper, we use the same problem of clock skew minimization for 3-D ICs in [8] by controlling temperature profile to show applicability and convergence trend of TSBO. The problem is posed as a 5-D problem which includes the control of heat transfer coefficient, thermal interface material (TIM), TIM thickness, underfill material, and printed circuit board material with corresponding bounds shown in Fig. 7. For brevity, we refer readers to [8] for the detailed description of the problem along with the accuracy of the simulation framework as compared with measurement results.

Simulations used in this paper assume a random generated static power map on the chip, as in [8]. Temperature profiles are calculated using aforementioned electrical–thermal



(a)



(b)

Fig. 8. Performance of proposed algorithm for clock skew minimization. (a) Temperature gradient. (b) Resulting clock skew.

solver, which uses finite volume method to simultaneously solve voltage distribution and thermal equations considering temperature-dependent electrical resistivity and Joule heating effect [23]. The solver uses 3-D nonuniform mesh and domain decomposition to capture the multiscale geometry of 3-D IC. Calculated temperature profiles are then fed into commercial circuit solvers to perform temperature-dependent electrical simulation for signal integrity analysis, which provides the clock skew.

The optimization objective is chosen as the minimization of temperature gradient which results in minimized clock skew. Convergence curves in Fig. 8 show the change in temperature gradient and clock skew with function queries, i.e., number of simulations. Previous work has shown that IMGPO and a nonlinear solver outperform other non-ML algorithms, including pattern search, genetic algorithm, and multistart method [8]. To make a direct comparison, we use two best algorithms considered in [8] and compare performance of TSBO with nonlinear solver (*fmincon* in MATLAB) and IMGPO. Optimization using TSBO resulted in a temperature gradient of 23.5 °C and clock skew of 86.0 ps, compared with 23.8 °C and 88.0 ps presented as preceding work with IMGPO and 25.2 °C and 96.6 ps with a nonlinear solver, which corresponds to an improvement of 4.7% and 2.3% as compared with IMGPO and 9.4% and 12.3% as compared with a nonlinear solver. The most compelling contribution of the algorithm is the significant decrease in CPU time required for convergence where TSBO converged to minimum temperature gradient 3.76 and 3.96 times faster than nonlinear solver

TABLE III
OPTIMIZATION RESULTS FOR 3-D IC

	Non-Linear Solver	Previous Work [8]	This Work
T_{GRAD} [°C]	25.2 (+9.4%)	23.8 (+4.7%)	23.5
Skew [ps]	96.6 (+12.3%)	88.0 (+2.3%)	86.0
CPU Time (Normalized)	3.96	3.76	1.00

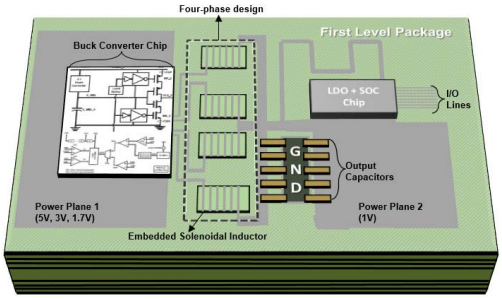


Fig. 9. Two-chip SiP IVR Architecture.

and previous work, respectively. The results of optimization for the three different algorithms are provided in Table III. Note that the normalized CPU times provided are calculated as the combined time of system simulations and algorithm progression time to converge to their corresponding minimum skews.

V. APPLICATION II: INTEGRATED VOLTAGE REGULATOR

The second application chosen in this paper is the multi-objective optimization of IVR with the goal of maximizing power efficiency and minimizing area of embedded inductor. The architecture used is given in Fig. 9 and is a System-in-Package (SiP) solution consisting of two chips (buck converter and LDO/load) with integrated inductor on an organic package [24]. The inductor structure is chosen as a solenoid with magnetic core as in Fig. 10(a) and (b). Due to high levels of integration, maintaining high conversion efficiency while minimizing the area of inductor in the package becomes a major challenge that requires the handling of multiple tradeoffs simultaneously.

In this section, we use TSBO to automate the design process. We briefly provide previously developed buck converter efficiency model followed by the embedded inductor characterization using two commercial EM solvers, namely Ansys HFSS [25] and Ansys Maxwell [26] for full-wave simulations at high frequency and at dc, respectively. Afterward, we provide the optimization setup used and make a comparison of the optimized IVR design with hand-tuned design [24] as well as results generated using other non-ML and ML techniques, such as a nonlinear solver, GP-UCB, and IMGPO.

A. Buck Converter Efficiency Model

The buck converter shown in Fig. 9 is designed with the stacked topology, using 130-nm GF process and consists of four phases (one master and three slaves) [24]. Efficiency calculations are based on the extensive model that accounts for switching and conduction losses of power switches, dc and ac losses of inductor, power delivery network (PDN), and output capacitance. Power switches used in the buck converter introduces two types of losses, namely switching and conduction. Conduction loss originates from the finite dc resistance of switches and can be written as

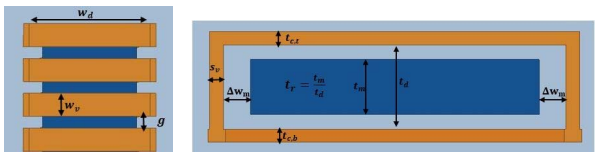


Fig. 10. Geometry of solenoidal inductor with magnetic core. (a) Top view. (b) Side view.

$$L_{PS(COND)} = R_{SW}(I^2_{LOAD} + \Delta I^2_{RMS}) \quad (18)$$

$$R_{SW} = D_{CCM} R_{PMOS} + (1 - D_{CCM}) R_{NMOS} \quad (19)$$

where R_{NMOS} and R_{PMOS} are ON-resistances of NMOS and PMOS, respectively, and ΔI_{RMS} is the *RMS* value of ripple current calculated as

$$D_{CCM} = V_{OUT}/V_{IN} \quad (20)$$

$$\Delta I_{pk-pk} = \frac{(V_{IN} - V_{OUT}) D_{CCM}}{f_{SW} L} \quad (21)$$

$$\Delta I_{RMS} = \Delta I_{pk-pk}/2\sqrt{3} \quad (22)$$

where D_{CCM} is duty cycle in continuous conduction mode.

The switching loss of power switches is given as

$$L_{PS(SW)} = f_{SW}\big[(C_{GSN} + C_{GSP})(V_{IN}/2)^2 \cdots + 2(C_{GDN} + C_{GDP})(V_{IN}/2)^2\big] \quad (23)$$

where C_{GSP}, C_{GSN}, C_{GDP}, and C_{GDN} are gate-to-source and gate-to-drain capacitances of PMOS and NMOS, respectively. The contribution of the inductor in terms of ac and dc resistances can be written as

$$L_{IND,dc} = I^2_{LOAD} R_{IND,dc} \quad (24)$$

$$L_{IND,ac} = \sum_{f=-\infty}^{\infty} 2x \frac{\Delta I^2_{pk-pk}}{4} [0.405^2 x \mathrm{ESR}_L(f_{SW}) \cdots + 0.045^2 x \mathrm{ESR}_L(3 f_{SW}) + 0.016^2 x \mathrm{ESR}_L(5 f_{SW})] \quad (25)$$

where $R_{IND,dc}$ and effective series resistance (ESR_L) are dc and frequency-dependent effective series resistance of an inductor, respectively; 0.405, 0.045, and 0.016 are Fourier series coefficients of fundamental frequency and its third and fifth harmonics of triangular current waveform. Resistive losses due to output capacitor and PDN are given as

$$L_{CAP} = \Delta I^2_{pk-pk} \mathrm{ESR}_C \quad (26)$$

$$L_{PDN} = I^2_{LOAD} R_{PDN} \quad (27)$$

where R_{PDN} is the effective dc resistance of PDN and ESR_C is effective series resistance of output capacitance. A more

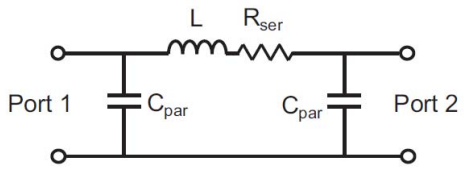


Fig. 11. π-equivalent inductor model.

detailed description of the buck converter topology along with the model verification can be found in [24].

B. Embedded Inductor Characterization

Although there are approximate closed-form expressions for preliminary design of embedded solenoidal inductors with magnetic core [27], a full-wave simulation is required to accurately account for eddy currents, proximity, and skin effect as well as demagnetization effect due to using magnetic cores.

Electrical characteristics of inductor have both direct and implied effects to the efficiency of the buck converter. The inductance needs to be sufficiently high in order to reduce the ripple current in (21), thereby reducing the conduction loss in (19). The ripple current also affects the ac loss in inductor along with ESR in (25). Although dc resistances of such inductors are in the range of milliohms, operating at higher currents ($\sim$10 A) introduces a substantial dc loss as in (24). Therefore, the complexity of the problem includes handling inductance, ac and dc resistance tradeoffs, and area constraints, while considering the direct and implied effects on buck converter efficiency. Even if the desired characteristics of the inductor is determined to handle these tradeoffs, determining the dimensions of the inductor that will satisfy these characteristics requires substantial human intervention and CPU time due to the use of full-wave EM solvers.

In order to characterize the inductor behavior, the full-wave EM simulation is conducted to gather two-port impedance matrix as a function of frequency. Using a pi-equivalent circuit as in Fig. 11, the inductance and ESR are calculated from the Z-matrix as

$$L = \frac{2\mathrm{Im}\{Z_{11} - Z_{12}\}}{w}, \quad \mathrm{ESR} = 2\mathrm{Re}\{Z_{11} - Z_{12}\}. \tag{28}$$

Finally, Ansys Maxwell [26] is used to accurately simulate the dc resistance of inductor to be used in (24) of the efficiency model.

In this paper, we consider two different magnetic materials, namely carbonyl iron powder (CIP) and nickel–zinc (NiZn). Between the two materials, NiZn has higher permeability ($\mu_r = 8.11 + j2.27$ at 100 MHz), which provides higher inductance values at reduced sizes, which decreases ripple current and dc resistance simultaneously and increases the efficiency. On the other hand, CIP has lower magnetic loss tangent ($\mu_r = 5.64 + j0.57$ at 100 MHz), which results in increased efficiency by reducing ac loss in (25). To accurately account for magnetic material effect on IVR efficiency, measured frequency-dependent complex permittivity and permeability of both CIP and NiZn from [28] are imported into the EM solver for accuracy.

TABLE IV
CONTROL PARAMETERS OF SOLENOIDAL INDUCTOR

Parameter		Unit	Min	Max
Gap between windings	g	mil	2	20
Number of windings	N		3	13
Size of via	s_v	μm	50	103
Copper Trace Width	w_c	mil	2	20
Copper Thickness Bottom	$t_{c,b}$	μm	35	170
Copper Thickness Top	$t_{c,t}$	μm	35	170
Dielectric Thickness	t_d	μm	50	650
Dielectric Width	w_d	mil	50	350
Magnetic Core Thickness Ratio	t_m		0.1	1
Magnetic Core Width offset	Δw_m	mil	0	100

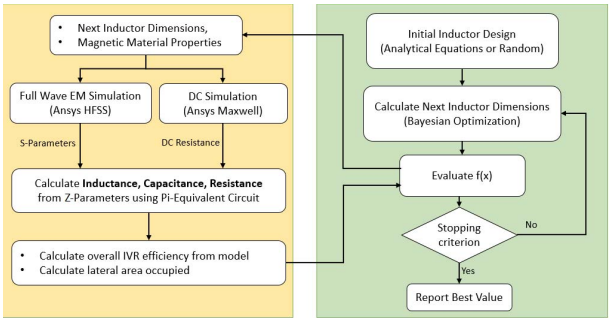


Fig. 12. Automated optimization setup used in IVR application.

C. Optimization Setup

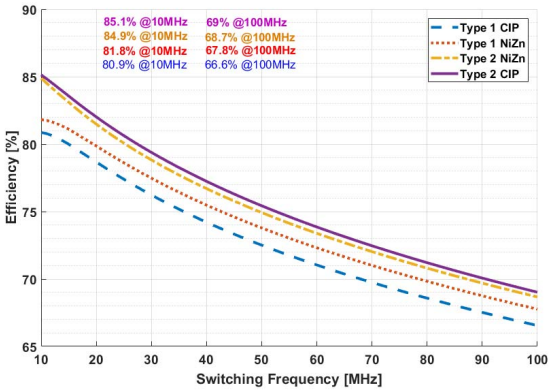
Among the components of the IVR, the focus in this paper is on the optimization of inductor to maximize IVR efficiency. Ten control parameters are defined for the inductor as shown in Table IV along with the range used for each parameter based on process capabilities. The objective function used for optimization is

$$f(x) = \sum_{i=1}^{2} w_i y_i \tag{29}$$

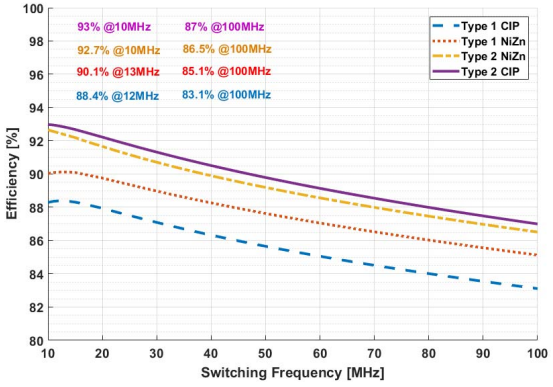
where y_1 and y_2 are peak overall IVR efficiency calculated using the described model and area of inductor, respectively; $w_1 = 5$ and $w_2 = -2$ are corresponding weights of multiobjective optimization. Since IVR efficiency is the main goal of the optimization process, it has higher weight compared with the inductor area. By choosing such weights, TSBO adopts the control parameters providing 2% of power efficiency over 5 mm^2 of the inductor area.

To investigate opportunities for reducing fabrication costs, two types of inductors are analyzed: one that uses only 1 oz copper (Type I) and the more costlier option that uses copper thickness up to 170 μm (Type II). In the case of Type II inductor, the optimization is done with ten parameters, and for Type I, only eight parameters are used, since $t_{c,b}$ and $t_{c,t}$ are fixed to 35 μm [29].

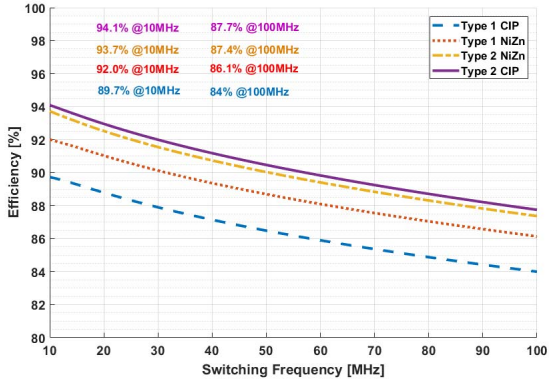
The automated optimization setup used is given in Fig. 12. Inductor dimensions are determined using TSBO and fed into the full-wave solver to extract the two-port Z-matrix, which is then used by the inductor and buck converter models for



(a)



(b)



(c)

Fig. 13. Efficiency comparison of four optimized IVRs as a function of switching frequency for a total $I_{\text{Load}} = 10$ A (2.5 A per phase). (a) 5 V:1 V conversion. (b) 3 V:1 V conversion. (c) 1.7 V:1.05 V conversion.

calculating IVR efficiency. Calculated efficiency is combined with inductor area in (29) and fed back into TSBO to proceed to the next iteration.

D. Results

To make a direct comparison with hand-tuned design in [24], power efficiencies are calculated assuming $R_{\text{PDN}} = 10\ \text{m}\Omega$ and $\text{ESR}_C = 10\ \text{m}\Omega$ in (27) and (26). Among four optimized IVRs, the one using Type II inductor with CIP showed the best performance, providing efficiency of 85.1%, 93.1%, and 94.1% for 5V:1V, 3V:1V and 1.7V:1.05V conversions, respectively, with an inductor area of 5.1 mm^2. Compared with hand-tuned design in [24], the efficiency is increased by 5.7%, 4.5%, and 3% accompanied with 55.3% reduction in inductor area.

TABLE V
COMPARISON OF OPTIMIZED IVRS AND INDUCTORS TO HAND-TUNED DESIGN

	Hand Tuned [24]	Type I NiZn	Type I CIP	Type II NiZn	Type II CIP
L [nH]	24.8	13.32	15.4	29.4	23.47
R_{AC} [Ω]	2.27	0.87	0.51	2.67	0.98
R_{DC} [mΩ]	14.7	17.0	30.1	10.5	8.7
Area [mm^2]	11.6	6.0	6.1	5.2	5.1
Peak eff. 5V:1V	79.4%	81.8%	80.9%	84.9%	85.1%
Peak eff. 3V:1V	88.5%	88.4%	90.1%	92.7%	93.0%
Peak eff. 1.7V:1.05V	91.1%	92.0%	89.7%	93.7%	94.1%

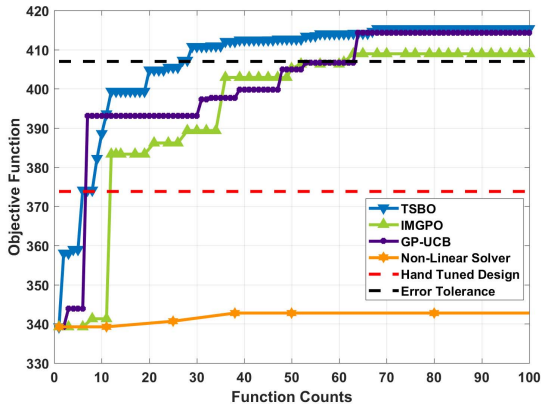
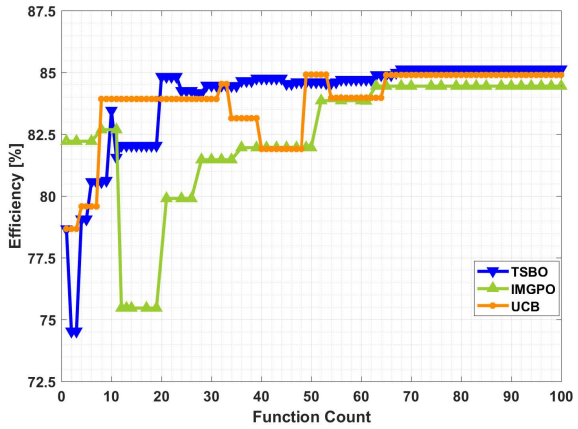


Fig. 14. Performance comparison of TSBO to nonlinear solver, GP-UCB, and IMGPO on maximizing objective function in (29).

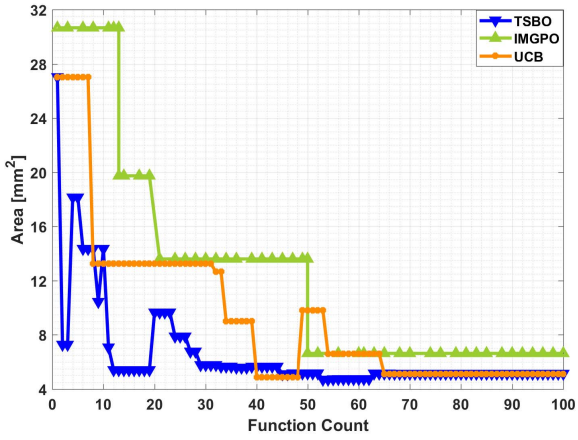
On the other hand, the cheaper option IVRs with Type I inductors showed comparable efficiency with hand-tuned design but with 48.2% reduced inductor area. In this case, IVR using inductor with NiZn core outperformed the inductor with CIP. This shows that when copper thickness is limited, using materials with higher permeability as compared with lower magnetic loss tangent provides more opportunities to increase IVR efficiency by decreasing number of windings and hence dc loss in (24).

Comparison for power efficiencies of each optimized IVR as a function of switching frequency at $I_{\text{LOAD}} = 10$ A (2.5 A per phase) is given in Fig. 13. In addition, Table V shows the inductor characteristics of each optimized IVR as well as peak conversion efficiencies and compares it with the hand-tuned design from [24].

Fig. 14 compares the performance of TSBO with a nonlinear solver, GP-UCB, and IMGPO for the objective of maximizing (29), along with the corresponding value of the hand-tuned design. Optimization using TSBO resulted in 85.1% peak efficiency for 5V:1V conversion with the inductor area



(a)



(b)

Fig. 15. Convergence comparison for the two components of objective function (29). (a) Peak 5 V:1 V efficiency at $I_{\mathrm{LOAD}} = 10$ A. (b) Inductor area.

TABLE VI
OPTIMIZATION RESULTS FOR IVR USING TYPE II INDUCTOR WITH CIP

	Non-Linear Solver	GP-UCB	IMGPO	TSBO
Peak eff. 5V:1V (f_{SW}=10MHz, I_{Load}=10A)	78.6 %	84.9 %	84.4 %	85.1 %
Area	25.2 mm^2 (+79.6%)	5.18 mm^2 (+0.4%)	6.64 mm^2 (+22.3%)	5.16 mm^2
CPU Time	> 185 min (> +72.9%)	117.3 min (+57.4%)	115.6 min (+56.7%)	50.1 min

of 5.16 mm^2, compared with 78.6%, 84.4%, and 84.9% with 25.2, 6.64, and 5.18 mm^2 for a nonlinear solver, IMGPO, and GP-UCB, respectively. Though all algorithms started from the same initial point, TSBO reached the predetermined error tolerance in Fig. 14 using 27 simulations (51.1 min), compared with 60 and 59 simulations (115.6 and 117.3 min) for IMGPO and GP-UCB, corresponding to a reduction of 56.7% and 57.4% in CPU time. Besides, optimization using a nonlinear solver resulted in 78.6% peak efficiency with the inductor area of 25.2 mm^2 and could not reach the error tolerance within 100 simulations (185 min).

Fig. 15 shows the breakdown of objective function in (29) to its two components of peak efficiency and inductor area. This breakdown shows that the TSBO converges around 30 simulations resulting in 84.5% efficiency with 5.78-mm^2 area, whereas IMGPO and GP-UCB converge around 60 simulations. This shows that the number of simulations required for each algorithm to cross the tolerance level in Fig. 14 coincides with their convergence point and hence validates the CPU time comparison made in Table VI.

VI. CONCLUSION

In this paper, we have proposed a new BO-based global optimization algorithm, TSBO, that considers EDA-related challenges in optimization. Unlike conventional BO algorithms, the proposed algorithm is not prone to initial point selection and does not rely on other algorithms for auxiliary optimization. Furthermore, we have proposed and used a new hierarchical partitioning scheme that makes the algorithm EDA-oriented in terms of substantially reducing the number of simulations required to reach the global optima. The empirical analysis on a set of popular challenge functions with several local extrema and dimensions showed TSBO to have a faster convergence trend over other widely used methods.

Furthermore, we have shown how ML, in particular, TSBO enables enhanced designs for two emerging applications, namely 3-D ICs and IVRs. In the IVR application, optimized IVR with Type II inductor with CIP using TSBO resulted in peak IVR efficiency for 5 V:1 V (at $I_{\mathrm{LOAD}} = 10$ A and $f_{\mathrm{SW}} = 10$ MHz) of 85.1% with the embedded solenoidal inductor occupying an area of 5.1 mm^2 corresponding to 5.7% increase in efficiency and 56.1% reduction of area compared with the hand-tuned design. Moreover, TSBO showed 72.4%, 57.4%, and 56.7% reduction in CPU time required to complete optimization compared with the nonlinear solver, GP-UCB, and IMGPO, respectively. This trend of faster convergence of TSBO was also observed in application for clock skew minimization of 3-D ICs, where TSBO was shown to be 3.76 and 3.96 times faster than IMGPO and nonlinear solver. The results presented in this paper show the proposed algorithms applicability to a variety of systems that can be represented as a black box, and hence the proposed method is very general.

REFERENCES

[1] W. R. Davis *et al.*, "Demystifying 3D ICs: The pros and cons of going vertical," *IEEE Design Test Comput.*, vol. 22, no. 6, pp. 498–510, Nov. 2005.
[2] G. H. Loh, Y. Xie, and B. Black, "Processor design in 3D die-stacking technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, May/Jun. 2007.
[3] N. Sturcken *et al.*, "A 2.5 D integrated voltage regulator using coupled-magnetic-core inductors on silicon interposer," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 244–254, Jan. 2013.
[4] N. Sturcken *et al.*, "A switched-inductor integrated voltage regulator with nonlinear feedback and network-on-chip load in 45 nm SOI," *IEEE J. Solid-State Circuits*, vol. 47, no. 8, pp. 1935–1945, Aug. 2012.
[5] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.
[6] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. (2009). "Gaussian process optimization in the bandit setting: No regret and experimental design." [Online]. Available: https://arxiv.org/abs/0912.3995
[7] P. Chen, B. M. Merrick, and T. J. Brazil, "Bayesian optimization for broadband high-efficiency power amplifier designs," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4263–4272, Dec. 2015.

[8] S. J. Park, B. Bae, J. Kim, and M. Swaminathan, "Application of machine learning for optimization of 3-D integrated circuits and systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 6, pp. 1856–1865, Jun. 2017.

[9] H. M. Torun and M. Swaminathan, "Black box optimization of 3D integrated systems," in *Proc. Comput. Modelling Multi Uncertainty Multi-Scale Problems*, Sep. 2017.

[10] H. M. Torun and M. Swaminathan, "Black box optimization of 3D integrated systems using machine learning," in *Proc. IEEE 26th Conf. Elect. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2017.

[11] R. M. Neal, *Bayesian Learning for Neural Networks* (Lecutre Nots in Statistics), vol. 118. Springer, 1996.

[12] B. Sriperumbudur, K. Fukumizu, and G. Lanckriet, "On the relation between universality, characteristic kernels and RKHS embedding of measures," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, 2010, pp. 773–780.

[13] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, vol. 1. Cambridge, MA, USA: MIT Press. 2006.

[14] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *Proc. ICML*, 2014, pp. 937–945.

[15] E. Brochu, V. M. Cora, and N. de Freitas. (Dec. 2010). "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning." [Online]. Available: https://arxiv.org/abs/1012.2599

[16] K. Kawaguchi, L. P. Kaelbling, and T. Lozano-Pérez, "Bayesian optimization with exponential convergence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2809–2817.

[17] Z. Wang, B. Shakibi, L. Jin, and N. Freitas, "Bayesian multi-scale optimistic optimization," in *Proc. Artif. Intell. Stat.*, 2014, pp. 1005–1014.

[18] R. Munos, "Optimistic optimization of a deterministic function without the knowledge of its smoothness," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 783–791.

[19] M. D. Hoffman, E. Brochu, and N. de Freitas, "Portfolio allocation for bayesian optimization," in *Proc. UAI*, 2011, pp. 327–336.

[20] S. Surjanovic. *Virtual Library of Simulation Experiments*. Accessed: Jan. 4, 2018. [Online]. Available: https://www.sfu.ca/~ssurjano/

[21] D. J. Lizotte, "Practical Bayesian optimization," Ph.D. dissertation, Dept. Comput. Sci., Univ. Alberta, Edmonton, AB, Canada, 2008.

[22] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *J. Global Optim.*, vol. 21, no. 4, pp. 345–383, 2001.

[23] J. Xie and M. Swaminathan, "Electrical-thermal co-simulation of 3D integrated systems with micro-fluidic cooling and Joule heating effects," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 1, no. 2, pp. 234–246, Feb. 2011.

[24] S. Mueller *et al.*, "Design of high efficiency integrated voltage regulators with embedded magnetic core inductors," in *Proc. IEEE 66th Electron. Compon. Technol. Conf. (ECTC)*, May 2016, pp. 566–573.

[25] ANSYS. *ANSYS HFSS Ver. 2015.2*. Accessed: Jan. 4, 2018. [Online]. Available: http://www.ansys.com

[26] ANSYS. *ANSYS Maxwell Ver. 2015.2*. Accessed: Jan. 4, 2018. [Online]. Available: http://www.ansys.com

[27] D. S. Gardner, G. Schrom, F. Paillet, B. Jamieson, T. Karnik, and S. Borkar, "Review of on-chip inductor structures with magnetic films," *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 4760–4766, Oct. 2009.

[28] M. L. F. Bellaredj, S. Mueller, A. K. Davis, P. Kohl, M. Swaminathan, and Y. Mano, "Fabrication, characterization and comparison of FR4-compatible composite magnetic materials for high efficiency integrated voltage regulators with embedded magnetic core micro-inductors," in *Proc. IEEE 67th Electron. Compon. Technol. Conf. (ECTC)*, May/Jun. 2017, pp. 2008–2014.

[29] M. Swaminathan, "Power delivery for electronic system consortium (PDES)," Georgia Inst. Technol., Atlanta, GA, USA, 2017.

Hakki Mert Torun (S'15) received the B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2016. He is currently working toward the Ph.D. degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, under the supervision of Dr. M. Swaminathan.

His current research interests include developing machine learning algorithms for system optimization and modeling with the applications in VLSI, power electronics, high-speed channels, and microwave systems.

Madhavan Swaminathan (M'95–SM'98–F'06) received the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1989 and 1991, respectively.

He was with IBM, where he was involved in packaging for supercomputers. He held the position of the Joseph M. Pettit Professor in Electronics in Electrical and Computer Engineering and the Deputy Director of the NSF Microsystems Packaging Research Center, Georgia Institute of Technology, Atlanta, GA, USA, where he is currently the John Pippin Chair in microsystems packaging and electromagnetics with the School of Electrical and Computer Engineering and the Director of the Center for Codesign of Chip, Package, System. He is the Founder and Co-Founder of two startup companies, namely E-System Design and Jacket Micro Devices. He has authored over 450 refereed technical publications and authored and co-edited three books. He holds 30 patents.

Dr. Swaminathan is the Founder of the IEEE Conference Electrical Design of Advanced Packaging and Systems, a premier conference sponsored by the European Physical Society on Signal Integrity in the Asian Region. He has served as the Distinguished Lecturer for the IEEE EMC Society.

Anto Kavungal Davis received the B.Tech. degree in electrical and electronics engineering from the National Institute of Technology at Tiruchirappalli, Tiruchirappalli, India, in 2006 and the M.Tech. and Ph.D. degrees in electronics design from the Indian Institute of Science, Bengaluru, India, in 2010 and 2015, respectively. During his Ph.D., he focused on noise reduction techniques for microprocessor power distribution networks and methods to suppress antiresonance peaks.

He was with Huawei Technologies from 2006 to 2007 and Brocade Communications Systems, Inc., in 2011. He is currently a Post-doctoral Researcher at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. He is currently involved in integrated voltage regulators and RF near-field coupled wireless power transfer system with switching regulators for the Internet of Things (IoT). His current research interests include electromagnetic compatibility, power integrity, switched capacitor converters, switched inductor converters, wireless power transfer, and IoT.

Mohamed Lamine Faycal Bellaredj received the Engineering degree in electronics and the Magister degree in physics from the Oran University of Science and Technology-Mohamed Boudiaf, Bir El Djir, Algeria, in 2005 and 2007, respectively, and the Ph.D. degree in microengineering from the University of Franche-Comt, Besançon, France, in 2013.

His current research interests include microengineering/nanoengineering, MEMS/NEMS modeling, fabrication and characterization, the physics and technology of semiconductor devices, and third generation solar cells.