Multi-objective Bayesian Optimization for Analog/RF Circuit Synthesis

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

ABSTRACT

In this paper, a novel multi-objective Bayesian optimization method is proposed for the sizing of analog/RF circuits. The proposed approach follows the framework of Bayesian optimization to balance the exploitation and exploration. Gaussian processes (GP) are used as the online surrogate models for the multiple objective functions. The lower confidence bound (LCB) functions are taken as the acquisition functions to select the data point with best Pareto-dominance and diversity. A modified non-dominated sorting based evolutionary multi-objective algorithm is proposed to find the Pareto Front (PF) of the multiple LCB functions, and the next simulation point is chosen from the PF of the multiple LCB functions. Compared with the multi-objective evolutionary algorithms (MOEA) and the state-of-the-art online surrogate model based circuit optimization method, our method can better approximate the Pareto Front while significantly reduce the number of circuit simulations.

CCS CONCEPTS

• **Hardware → Analog and mixed-signal circuit optimization**;

ACM Reference format:
Wenlong Lyu[1], Fan Yang[1,*], Changhao Yan[1], Dian Zhou[1,2], Xuan Zeng[1,*] . 2018. Multi-objective Bayesian Optimization for Analog/RF Circuit Synthesis. In *Proceedings of DAC '18: The 55th Annual Design Automation Conference 2018, San Francisco, CA, USA, June 24–29, 2018 (DAC '18),* 6 pages.
DOI: 10.1145/3195970.3196078

1 INTRODUCTION

The scaling IC technology and increasing demands of high-performance, low-power circuits make the manual analog IC design much more difficult. Automated analog circuit sizing has attracted more research interests [19]. The analog circuit sizing can be well formulated as nonlinear optimization problems. The well-developed optimization algorithms and the ever-increasing computational capacity make the solving of the optimization problems tractable.

Most of the analog circuit sizing methods treat the sizing problem as a single-objective optimization problem. However, one circuit could have several conflicting performances, and it is usually impossible to simultaneously optimize all of them. One intuitive and widely adopted approach is to incorporate the designs' prior knowledge and assign each performance a weight to compose a weighted cost function. The problem with this approach is that it is difficult to assign appropriate weights for the performances. Wrong weights might lead to inappropriate designs.

The multi-objective optimization provides an alternative perspective to the sizing problems. Instead of optimizing a single objective and seeking for the single best design, multi-objective optimization methods try to find a set of uniformly distributed *Pareto-optimal* design to approximate the *Pareto Front(PF)* [15], which represents the most ambitious trade-off of the conflicting performances. The approximation of PF is presented to the designer for further decision. Also, the cell-level PFs can serve as the behavior models of cells and be used in hierarchical system-level optimizations [6].

Similar to single-objective analog circuit sizing, most of the existing multi-objective sizing methods can be classified as metamodel-based and simulation-based approaches. Metamodel-based multi-objective optimization methods model the circuit performances as analytical functions. The multi-objective optimization is then performed based on these computationally cheap models. For example, the geometric programming (GP) and posynomial modeling are combined with multi-objective evolutionary algorithms to handle the multi-objective sizing problems in [10, 18]. On the other hand, in the simulation-based methods, the optimization is driven directly by the circuit simulations. The multiple objective functions are viewed as black-box functions. They are obtained directly from circuit simulations. Meta-heuristic based algorithms are widely adopted. For example, the non-dominated sorting genetic algorithm (NSGA-II) [5] is used in [12, 16]. In [13], the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [25] serves as the optimization kernel. The scalarization scheme, which transforms the multi-objective problem into a sequence of single-objective problems is adopted in [21]. In [6], the PF of the system is generated via reuse of PFs of subblocks, while the PFs of subblocks are still generated from the evolutionary algorithm.

The accurate models are crucial to the model-based methods. However, the accurate models are hard to manually derive or need a lot of simulation data to fit. The simulation-based methods may also need a large number of simulations. If the circuit simulations are computationally expensive, these approaches can be very time-consuming. To address this problem, a hybrid methodology which combines both the model-based and simulation-based approaches has been proposed recently. Instead of using pre-built offline models, online evolving models are built. The circuit simulations are invoked during the optimization procedure to update the online models incrementally. The online surrogate model-based method GPMOOG was proposed in [11]. The GPMOOG method [11] uses the Gaussian process (GP) as the surrogate model. The evolutionary multi-objective optimization algorithm MOEA/D [9, 25] is used as the optimization kernel. The GP model provides the predictions of the performances as well as their uncertainties. In each generation of the MOEA/D evolution, the GP models are used to predict the objective functions of the children population. For the GP predictions with large uncertainties, the Electro-Magnetic (EM) simulations are invoked for this generation of children. Otherwise, the GP predictions are used to replace the EM simulations. The GP models are refined while new EM simulation results are available.

In this paper, we propose a multi-objective Bayesian optimization algorithm (MOBO). Bayesian optimization [20] was proposed for the optimization of black-box functions that are computationally expensive to evaluate. In Bayesian optimization, the Gaussian process

[*] Corresponding authors: yangfan@fudan.edu.cn, xzeng@fudan.edu.cn

DAC '18, San Francisco, CA, USA
© 2018 ACM. 978-1-4503-5700-5/18/06...$15.00
DOI: 10.1145/3195970.3196078

model is used as the surrogate model. In contrast to GPMOOG [11], where the uncertainties predicted by GP models are used only to decide whether the EM simulation should be invoked, the GP model participates more deeply in the optimization progress. Acquisition functions are constructed from the predictions and their uncertainties provided by GP models. The acquisition function is used to balance the *exploitation* and *exploration* during the optimization procedure. It tends to visit the optimal points which are predicted by the surrogate model with high probability (exploitation), as well as the regions with high uncertainties (exploration). In this work, the single-objective Bayesian optimization is extended to handle multi-objective optimization, the lower confidence bound (LCB) function is chosen as the acquisition function for each objective. We proposed a non-dominated sorting based evolutionary multi-objective algorithm for the multi-objective optimization of the multiple LCB functions. After the Pareto Front (PF) of the multiple LCB functions are obtained, a new data point is selected from the obtained PF via the modified non-dominated sorting. The performances of this new data point are evaluated with circuit simulations and are then used to refine the GP models. Two benchmark functions and three real-world examples are used to test the MOBO algorithm, compared with the MOEA/D and GPMOOG algorithm, the proposed approach can better approximate the Pareto Front while significantly reduce the number of circuit simulations.

The rest of the paper is organized as follows. In Section 2, the background of the multi-objective optimization will be reviewed. In Section 3, the proposed multi-objective Bayesian optimization algorithm MOBO will be presented. The experimental results are demonstrated in Section 4. We conclude the paper in Section 5.

2 BACKGROUND

In this section, we will give the formulation of the multi-objective optimization, and briefly review the Gaussian Process surrogate model.

2.1 Multi-Objective Optimization

The multi-objective objective optimization can be formulated as

$$\text{minimize } f_1(\boldsymbol{x}),\ \ldots\ , f_m(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \in \mathrm{D} \subset \mathrm{R}^d$ is the design variables and $f_i(\boldsymbol{x})$ represents the i-th cared circuit performance like gain or phase margin of an amplifier. Unlike single-objective problems, there is usually no single best design for a multi-objective problem. The goal of multi-objective optimization is to find a set of *Pareto-optimal* designs [15]. In multi-objective optimization, a design $\boldsymbol{x}_1$ is said to *dominate* $\boldsymbol{x}_2$, if $\forall i \in \{1 \ldots m\}$, $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ and $\exists j \in \{1 \ldots m\}$, such that $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$. A design is said to be *Pareto-optimal* if no design in the whole design space dominates it. The set of Pareto-optimal outcomes is called the *Pareto Front* (PF). Multi-objective optimization algorithm tries to find a set of evenly distributed Pareto-optimal designs to approximate the true Pareto front.

2.2 Gaussian Process Regression

Gaussian process (GP) regression model is used as the surrogate model in [11]. The advantage of GP is that it not only provides a prediction like other regression models but also gives a well-calibrated uncertainty estimation. The uncertainties would be small in the regions with densely sampled data and large in the regions with few samples.

GP is fully characterized by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{y})$. In this work, we use the constant mean function $m(\boldsymbol{x}) = \mu_0$ and the squared exponential covariance function as follows to characterize the Gaussian process.

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\Big(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \Lambda^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)\Big), \tag{2}$$

where $\Lambda = \operatorname{diag}(l_1, \ldots, l_d)$ is a diagonal matrix and l_i denotes the length scale of the i-th dimension. μ_0, σ_f and Λ are the hyper-parameters.

GP can be used to model a black-box function $f(x)$. Denote the hyper-parameters as a vector $\boldsymbol{\theta}$, given the training set $\{X, \boldsymbol{y}\}$, where $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{y} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))^T$. The log likelihood of the training set [17] can be expressed as

$$\log \mathrm{p}(\boldsymbol{y}|X, \boldsymbol{\theta}) \propto -\frac{1}{2}(\boldsymbol{y} - \mu_0)^T K_{\boldsymbol{\theta}}^{-1} (\boldsymbol{y} - \mu_0) - \frac{1}{2} \log |K_{\boldsymbol{\theta}}| \tag{3}$$

where $K_{\theta}(i, j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By maximizing the likelihood function (3), we can get the maximum likelihood estimation (MLE) of the hyper-parameters.

Given a new data point $\boldsymbol{x}$, the prediction of $f(\boldsymbol{x})$ is not a scalar value, but a Gaussian distribution

$$f(\boldsymbol{x}) \sim N(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})) \tag{4}$$

where $\mu(\boldsymbol{x})$ and $\sigma^2(\boldsymbol{x})$ can be expressed as

$$\left\{ \begin{array}{lcl} \mu(\boldsymbol{x}) & = & \mu_0 + k(\boldsymbol{x}, X) K_{\boldsymbol{\theta}}^{-1} (\boldsymbol{y} - \mu_0) \\ \sigma^2(\boldsymbol{x}) & = & \sigma_f^2 - k(\boldsymbol{x}, X) K_{\boldsymbol{\theta}}^{-1} k(X, \boldsymbol{x}) \end{array} \right. . \tag{5}$$

where $k(\boldsymbol{x}, X) = (k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_N))^T$ and $k(X, \boldsymbol{x}) = k(\boldsymbol{x}, X)^T$. The $\mu(\boldsymbol{x})$ can be viewed as the prediction of the function value, while the $\sigma^2(\boldsymbol{x})$ is a measure of uncertainty. We refer readers to [17] for more details of GP.

3 MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

In this section, we will present our proposed Multi-Objective Bayesian Optimization (MOBO) approach.

3.1 Bayesian Optimization

Bayesian optimization [20] was proposed for the optimization of expensive black-box functions. It consists of two essential ingredients, i.e., the probabilistic surrogate models and an acquisition function. The probabilistic surrogate models provide the prediction with uncertainties. They are refined incrementally with newly observed data. Acquisition function is used to explore the state space based on the surrogate model optimally.

In this subsection, we introduce the single-objective Bayesian optimization, our extension to multi-objective optimization would be described in the next subsection.

Without loss of generality, we only consider the minimization problem. For a black-box function $f(\boldsymbol{x})$, we aim to find the global minimum.

$$\boldsymbol{x}_* = \arg\min_{x} f(\boldsymbol{x}) \tag{6}$$

The GP model is the *de facto* surrogate model for Bayesian optimization. In Bayesian optimization, the black-box function $f(x)$ is modeled by the GP model (4). Through an acquisition function, the prediction and its uncertainty estimation can be combined to guide the optimization procedure efficiently.

In Bayesian optimization, the acquisition function is constructed to balance the exploration and exploitation. By optimizing the acquisition function, the next data point which can efficiently explore the state space. We consider an acquisition function lower confidence bound (LCB) [14] defined as follows.

$$\mathrm{LCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \kappa\sigma(\boldsymbol{x}) \tag{7}$$

where $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are derived from the GP prediction, and κ is a user-defined parameter, We set $\kappa = 3$ in this work. As can be seen, the prediction and uncertainty estimation are combined in the LCB function.

The LCB function would have a low value if $\mu(\boldsymbol{x})$ is low, or the uncertainty $\sigma(\boldsymbol{x})$ is high. It means that we tend to visit the data point where $f(x)$ is minimized based on the prediction and also the region where the prediction uncertainty is large.

Starting from a set of initial data points, in each iteration of Bayesian optimization, a GP model is trained using the existing data points. The acquisition function, e.g., the LCB function is then constructed, and existing optimization algorithms can be applied to optimize the acquisition function. The optimum of the acquisition function is then selected as the next data point to be evaluated. Note that, during the optimization of the acquisition function, only the evaluations of the GP models are involved, which are computationally efficient. In [4], it is proved that under some regularity assumptions, the LCB-based Bayesian optimization has exponentially vanishing instantaneous regret.

Besides LCB, there also exist other acquisition functions like expected improvement [1], Thompson sampling [2], and predictive entropy search [7]. A portfolio of several different acquisition functions is also possible [8].

3.2 Multi-Objective Bayesian Optimization

Bayesian optimization was initially proposed for single-objective optimization, however, with some simple modifications, we can easily extend it to handle multi-objective optimization. We still want a sequential algorithm such that in each iteration, one point is selected for circuit simulation. However, instead of defining a scalar-valued acquisition function, we construct an individual acquisition function for each objective's GP model, and perform multi-objective optimization to the multiple acquisition functions. A new point is then selected from the Pareto front of the acquisition functions.

For the multi-objective optimization problem (1), we construct LCB acquisition functions for all the objective functions.

$$\mathrm{LCB}_1(\boldsymbol{x}),\ \ldots\ ,\mathrm{LCB}_m(\boldsymbol{x}),$$

where $\mathrm{LCB}_i(\boldsymbol{x})$ denotes the LCB acquisition function for objective function $f_i(x)$ and

$$\mathrm{LCB}_i(\boldsymbol{x}) = \mu_i(\boldsymbol{x}) - \kappa_i \sigma_i(\boldsymbol{x}),$$

where $\mu_i(\boldsymbol{x})$ and $\sigma_i(\boldsymbol{x})$ are the mean and variance of the prediction of objective function $f_i(x)$ as shown in (5).

We select LCB as acquisition function based on the following considerations. Firstly, for the improvement based acquisition functions like expected improvement [1], a reference value is required to. For single-objective optimization, the reference can be defined as the optimal value of the objective function found currently. However, for multi-objective optimization, the reference is difficult to define, since we aim to explore the Pareto-optimal points, which are mutually non-dominated. Secondly, most of other acquisition functions like predictive entropy search [7] have no closed forms. LCB function has simple closed form and does not require a reference value.

Therefore, we consider solving the following multi-objective optimization problem.

$$\text{minimize } \mathrm{LCB}_1(\boldsymbol{x}),\ \ldots\ ,\mathrm{LCB}_m(\boldsymbol{x}). \tag{8}$$

Such a multi-objective optimization problem can be solved by existing multi-objective optimization algorithms. We propose a modified version of NSGA-II [5] to efficiently solve this problem, which will be discussed in next subsection. Note that the LCB functions $\mathrm{LCB}_1(\boldsymbol{x}),\ \ldots\ ,\mathrm{LCB}_m(\boldsymbol{x})$ have closed forms and are thus cheap to evaluate. Therefore, the computational cost to approximate their Pareto front would be relatively small. The point on the PF of the LCB functions can be viewed as good candidates of the next simulation point. If the GP models are accurate, the Pareto front of the LCB functions would be same with the Pareto front of the problem (1), if the GP models are still inaccurate, the Pareto-optimal points of the LCB functions will reside in areas where the uncertainties are high. After the Pareto front of the LCB functions is obtained, a simulation point is selected from the Pareto front, the selection mechanism is also described in the next subsection.

As the new data point is selected, the circuit simulation will be invoked to obtain the performances. The GP models are then updated with this new data point. This procedure will continue until the limit of the number of circuit simulations is reached, or other stop criteria are met. As the iteration proceeds, more data points near the PF of the original optimization problem (1) would be found, and the accuracy of the GP models near the Pareto front would thus be improved. Our solution will be closer to the true PF of the original optimization problem (1) gradually.

3.3 Modified NSGA-II for the Multi-Objective Optimization of the LCB Functions

We propose a modified NSGA-II algorithm for the multi-objective optimization of the LCB functions. The NSGA-II [5] algorithm defines a complete order between individual data points according to the dominance relationship and the distribution of points. Firstly, an integer rank is assigned to each point, given a set of points, the non-dominated points would be assigned as rank 0 and removed, the non-dominated points of the remaining points would then get a higher rank, this procedure recursively continues until the set is empty. If a point $\boldsymbol{x}_1$ is dominated by $\boldsymbol{x}_2$, then $\boldsymbol{x}_1$ would always have a higher rank value. Secondly, for a set of points with the same rank, a density measure called *crowding distance* is defined for each point. A point that is far away from its surrounding points would have a higher crowding distance. The concepts dominance rank and crowding distance are illustrated in Figure 1. In NSGA-II, the dominance rank and crowding distance form the fitness of the evolution together, a point with lower rank value and higher crowding distance is favored. With this fitness definition, the solutions can be compared and sorted, even if they do not dominate each other. In each iteration of NSGA-II, the children are generated from current population via mutation and crossover of the genetic algorithm, the parents and children are then combined and sorted using the non-dominated sorting, the top N points would be selected as the new population where N is the population size.

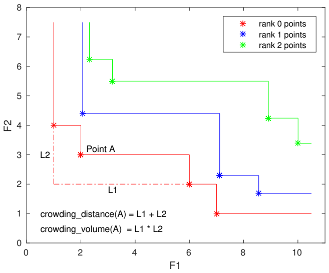

Figure 1: Illustration of non-dominated sorting

We make several modifications to the original NSGA-II. Firstly, the mutation and crossover operations of the genetic algorithm are replaced by the differential evolution (DE) [22] operations, this is inspired by [13] where the genetic operations of MOEA/D are also replaced by DE operations. Secondly, in the original definition of crowding distance, a solution's crowding distance of each objective is first calculated. The whole crowding distance is the summation of the crowding distances of all objectives. In our implementation, the multiplication instead of summation of the crowding distances of all objectives is used. This is because multiplication is scale-independent when used for comparison, so it doesn't require a normalization of different objectives. It can be viewed as the *crowding volume* rather than *crowding distance*. As shown in

Figure 1, for the point A, the crowding distance is $L1 + L2$, while the crowding volume is $L1 \times L2$. Thirdly, although we aim to solve the multi-objective optimization problem of the LCB functions, we think it would be useful to include all the simulated data points to guide the evolution, the already simulated data can be viewed as GP prediction with zero uncertainty. The simulated data points serve as "elitist anchor" in the evolution process, and are combined with the parents and children in each iteration of the evolutionary algorithm.

The modified non-dominated sorting is not only used *during* the multi-objective LCB optimization but also *after* the optimization. Once a set of Pareto-optimal points are found, we need to select a new data point from these points for the next circuit simulation. The non-dominated points are combined with the already simulated data, and a new round of non-dominated sorting is performed, the non-simulated point with the best fitness is chosen for the next simulation.

Note that any other existing multi-objective optimization algorithm can also be used in the MOBO framework for the multi-objective optimization of the LCB functions, as long as they find a good approximation of the PF of LCB functions.

4 EXPERIMENTAL RESULTS

In this section, we compared our MOBO algorithm with the simulation-based MOEA/D algorithm [13] and the GPMOOG algorithm [11] which also adopted the online surrogate model methodology. The ZDT1 and ZDT2 functions from the ZDT test benches [26] are taken as the first test case. The trues PF of ZDT1 and ZDT2 are known. The ZDT1 and ZDT2 functions can thus demonstrate the differences between the obtained PFs and the true PFs. Also, the ZDT functions are used to verify the correctness of our MOEA/D and GPMOOG implementations, as these two functions are also used in [11]. Then, three real-world examples are tested, including a three-stage operational amplifier, a transformer working at 60GHz and a power amplifier working at 2.5GHz. In all the experiments, we stop the algorithms while the number of simulation reach specified limits.

As the result of the multi-objective optimization is not a scalar value, there is no unique way to compare the performances of different multi-objective algorithms. Plenty of metrics have been proposed for the comparison [3]. In our experiments, the hypervolume difference [3] metric I_H^- is used to compare the performances of our proposed MOBO approach and the existing GPMOOG [11] and MOEA/D [13] approaches.

Given a set of non-dominated points P, the hypervolume of P is defined as the Lebesgue measure of the dominated space. As shown in Figure 2, for a 2-objective problem, the area of the shaded region is the hypervolume of the Pareto front. To calculate the hypervolume, a reference point is required, in this work, we combine all the sampled data to draw a reference point using the worst values of all the objectives. To have a high hypervolume value, the points in P must be close to the true Pareto front and can not miss any part of the true Pareto front.

Given a Pareto front approximation P, the I_H^- metric of P is defined as the difference between the hypervolumes of the true Pareto front P_* and P:

$$\mathrm{I}_{\mathrm{H}}^-(P) = \mathrm{HV}(P_*) - \mathrm{HV}(P) \tag{9}$$

For real-world problems, the true Pareto front P_* is unknown. In our experiments, all the sampled data are combined to extract a reference "true" Pareto front P_*. The HV(P_*) is a constant value for all the algorithms.

For each problem, each algorithm was independently repeated for ten times to average the random fluctuations. The statistics of I_H^- values are calculated. Also, we use two sample student-t test [23] to test whether the I_H^- values of GPMOOG and MOEA/D are different MOBO with *statistical significance*. The I_H^- values of MOBO are compared with GPMOOG and MOEA/D, and two corresponding p-values are calculated. A higher p-value implies that the difference between two sample is more likely due to the random variations. Generally, $p < 0.05$ or $p < 0.005$ are recognized as a significance level.

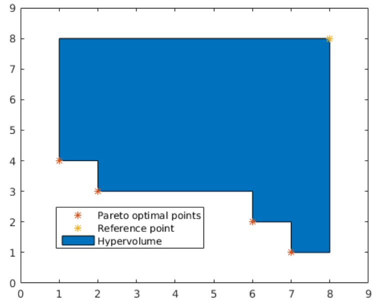

Figure 2: Illustration of hypervolume metric

4.1 ZDT1 and ZDT2 Functions

The ZDT1 and ZDT2 functions from the ZDT [26] benchmark problems are used in the first example. The number of decision variables is five. For our proposed MOBO method, we set the number of maximum of allowed evaluations to 300. For the other two algorithms, two different configurations are used. Firstly, we run the two algorithms with 600 maximum allowed evaluations. Then, we run GPMOOG with 1800 allowed evaluations and MOEA/D with 3600 allowed evaluations. Each configuration was repeated ten times. We plot the median results (by I_H^-) in Figure 3.

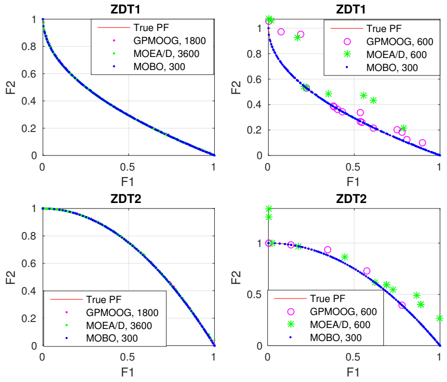

Figure 3: Optimization results of ZDT1 and ZDT2 functions

As can be seen, for our proposed MOBO algorithm, it can achieve very good approximations to the true Pareto fronts of ZDT1 and ZDT2 functions within only 300 evaluations. Although the GPMOOG and MOEA/D also achieved good results with 1800 and 3600 evaluations, they failed to generate acceptable Pareto fronts within 600 evaluations.

The results in Figure 3 also verifies the correctness of our GPMOOG and MOEA/D implementations. As in [11], the author optimized these two functions using the original GPMOOG and MOEA/D algorithms with 1800 and 6000 evaluations respectively. We refer readers to [11, Figure 6] where the median results are also plotted.

4.2 A Three-Stage Lower-Power Amplifier

The three-stage amplifier initially proposed in [24] is used as the second example. The amplifier is designed using SMIC 55nm process, it has 24 design variables, including the sizes of transistors, the capacitances, resistances, and the bias currents.

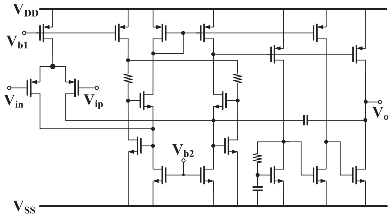

Figure 4: Schematic of the low power three-stage amplifier.

We consider five process corners (TT/FF/SS/SF/FS) and three temperatures (25°C/-40°C/125°C). A total of 15 simulations are required for one design. The worst performances across all corners are to be optimized. We aim to maximize the gain, unit gain frequency (UGF) and phase margin (PM) of this amplifier. The specifications are listed in (10).

$$\text{maximize } (Gain_c, UGF_c, PM_c) \tag{10}$$

where the $Gain_c$, UGF_c and PM_c are the minimum gain, UGF and PM values across all cared corners.

For this circuit, each run of our proposed MOBO approach consumes 400 simulations, including 80 initial random samples. 800 simulations are performed for GPMOOG and MOEA/D. The best objective values, statistics of the I_H^- values, and the p-values are listed in Table 1.

Table 1: Statistics of the amplifier optimization results

Algo	MOBO	GPMOOG	MOEA/D
Maximum Gain(dB)	**98.93**	86.76	83.38
Maximum UGF(MHz)	**4.03**	2.04	1.93
Maximum PM(°)	131.15	130.57	**132.62**
mean I_H^-	**398037**	435899	439516
median I_H^-	**396505**	442930	439610
max I_H^-	**432190**	456070	448410
min I_H^-	**359640**	390850	430370
p-value	—	0.0016	0.0001

As can be seen, with only half the number of simulations, our proposed MOBO approach found the best gain and UGF. Although the MOEA/D found a better PM value, the difference of best PM values of the three algorithms is negligible. As for the I_H^- metric, the MOBO algorithm achieved the smallest I_H^- values among the three algorithms, and the differences have very strong statistical significance.

4.3 A 60GHz Transformer

The layout of the transformer is shown in Figure 5. This transformer is designed with TSMC 65nm technology. ADS momentum is used for EM simulation in this example.

There are four design variables, including the radii and widths of metal-8, metal-9 layers, as listed in (11).

$$\begin{cases} R_{m8} & \in & [3, 75]\,\mu\text{m}, \\ R_{m9} & \in & [3, 75]\,\mu\text{m}, \\ W_{m8} & \in & [5, 10]\,\mu\text{m}, \\ W_{m9} & \in & [5, 10]\,\mu\text{m}. \end{cases} \tag{11}$$

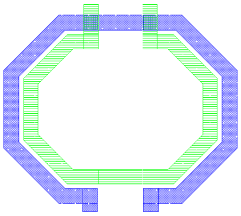

Figure 5: Layout of the transformer

The goal is to maximize the power transfer efficiency (PTE) while minimizing the area, as listed in (12).

$$\text{minimize} \qquad (-PTE,\ area) \tag{12}$$

We limit the number of simulations to 300 for our proposed MOBO approach, including 40 initial random samples. The numbers of simulations are limited to 350 for MOEA/D and GPMOOG methods. In Table 2, the best values of each objective are listed. The statistics of the I_H^- values and the p-values are also listed.

As can be seen, our proposed MOBO algorithm achieved much lower I_H^- metric value than GPMOOG and MOEA/D with extremely strong statistical significance. The best values of all the objectives of our proposed MOBO algorithm are also better than GPMOOG and MOEA/D.

Table 2: Statistics of the transformer optimization results

Algo	MOBO	GPMOOG	MOEA/D
Minimum Area(μm^2)	**64.27**	66.94	67.53
Maximum PTE(%)	**84.34**	82.33	82.29
mean I_H^-	**1379.63**	15863.9	18899
median I_H^-	**1443.03**	16475.9	20252
max I_H^-	**1766.34**	19392.4	22356
min I_H^-	**927.640**	10505.6	11664.4
p-value	—	2.43e-12	5.70e-12

The Pareto fronts with median I_H^- values are visualized in Figure 6. In Figure 6, the MOBO algorithm clearly generated the best PF, most points generated by GPMOOG and MOEA/D are actually dominated by MOBO.

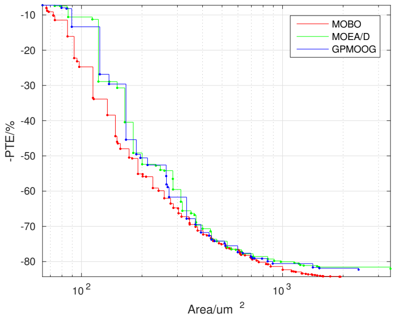

Figure 6: Pareto fronts generated by MOBO, GPMOOG and MOEA/D

4.4 A Power Amplifier

A power amplifier, working at 2.5GHz, designed using TSMC 65nm process is used as an example. The power amplifier is an array-based design, 2^{11} identical cells are duplicated. Each cell contains four transistors. There are five design variables, as shown in (13). This circuit is taken from a third-party corporation and is going to be taped out, so we are not allowed to post its schematic as it employs a new architecture.

$$\begin{cases} C_p & \in & [0.01, 20]\,\text{pF} \\ C_s & \in & [0.01, 20]\,\text{pF} \\ V_{dd} & \in & [1, 1.5]\,\text{V} \\ V_b & \in & [1, 2]\,\text{V} \\ W & \in & [0.12, 5]\,\mu\text{m} \end{cases} \tag{13}$$

For this power amplifier, we want to maximize its efficiency (Eff) and output power (Pout) while minimizing its non-linearity. The non-linearity is represented by the total harmonic distortion (Thd). The specifications are shown in (14).

$$\text{minimize } (-\mathit{Eff}, -\mathit{Pout}, \mathit{Thd}) \tag{14}$$

For our proposed MOBO approach, we limit the number of simulations to 300, including 40 initial random sampling. The numbers of simulations are limited to 500 for GPMOOG and MOEA/D. In Table 3, the best objective values, statistics of I_H^- values and the p-values are calculated. As can be seen, although our proposed MOBO algorithm consumed 40% fewer simulations than MOEA/D and GPMOOG, it found designs with best objective values, and its average I_H^- value is only about 20% of GPMOOG and MOEA/D. The I_H^- differences also have extremely strong significance.

Table 3: Statistics of the power amplifier optimization results

Algo	MOBO	GPMOOG	MOEA/D
Maximum Eff(%)	**65.12**	61.66	61.32
Minimum Thd(dB)	**5.90**	6.34	6.80
Maximum Pout(dBm)	**26.55**	25.27	25.30
mean I_H^-	**82860.1**	435116	466421
median I_H^-	**75233**	438670	437235
max I_H^-	**151350**	530570	624390
min I_H^-	**49411**	322380	285460
p-value	—	8.01e-11	3.78e-9

5 CONCLUSION

In this paper, a multi-objective Bayesian optimization (MOBO) algorithm is proposed for the multi-objective optimization of analog/RF circuits. Gaussian processes (GP) are used as the online surrogate models. The LCB functions are taken as the acquisition functions. A modified NSGA-II algorithm is proposed to find the Pareto Front (PF) of the multiple LCB functions. Compared with the state-of-the-art algorithms listed in this paper, the MOBO can generate better Pareto front with much fewer simulations.

ACKNOWLEDGMENTS

This research was supported partly by the National Major Science and Technology Special Project of China (2017ZX01028101- 003), partly by National Natural Science Foundation of China (NSFC) research projects 61774045,61574044, 61474026, 61574046, 61674042, and 61628402 and partly by the Recruitment Program of Global Experts (the Thousand Talents Plan).

REFERENCES

[1] Adam D Bull. 2011. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research* 12, Oct (2011), 2879–2904.

[2] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.

[3] Shi Cheng, Yuhui Shi, and Quande Qin. 2012. On the performance metrics of multiobjective optimization. *Advances in Swarm Intelligence* (2012), 504–512.

[4] Nando De Freitas, Alex Smola, and Masrour Zoghi. 2012. Exponential regret bounds for Gaussian process bandits with deterministic observations. *arXiv preprint arXiv:1206.6457* (2012).

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[6] Tom Eeckelaert, Trent McConaghy, and Georges Gielen. 2005. Efficient multiobjective synthesis of analog circuits using hierarchical pareto-optimal performance hypersurfaces. In *Proceedings of the conference on Design, Automation and Test in Europe-Volume 2*. IEEE Computer Society, 1070–1075.

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

[8] Matthew D Hoffman, Eric Brochu, and Nando de Freitas. 2011. Portfolio Allocation for Bayesian Optimization.. In *UAI*. 327–336.

[9] Hui Li and Qingfu Zhang. 2009. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE transactions on evolutionary computation* 13, 2 (2009), 284–302.

[10] Tuotian Liao and Lihong Zhang. 2017. Parasitic-aware GP-based many-objective sizing methodology for analog and RF integrated circuits. In *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 475–480.

[11] Bo Liu, Hadi Aliakbarian, Soheil Radiom, Guy AE Vandenbosch, and Georges Gielen. 2012. Efficient multi-objective synthesis for microwave components based on computational intelligence techniques. In *Proceedings of the 49th annual design automation conference*. ACM, 542–548.

[12] Bo Liu, Francisco V Fernandez, Peng Gao, and Georges Gielen. 2009. A fuzzy selection based constraint handling method for multi-objective optimization of analog cells. In *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*. IEEE, 611–614.

[13] B. Liu, F. V. FernÃąndez, Q. Zhang, M. Pak, S. Sipahi, and G. Gielen. 2010. An enhanced MOEA/D-DE and its application to multiobjective analog cell sizing. In *IEEE Congress on Evolutionary Computation*. 1–7. DOI:https://doi.org/10.1109/CEC.2010.5585957

[14] B. Liu, D. Zhao, P. Reynaert, and G. G. E. Gielen. 2014. GASPAD: A General and Efficient mm-Wave Integrated Circuit Synthesis Method Based on Surrogate Model Assisted Evolutionary Algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33, 2 (Feb 2014), 169–182. DOI:https://doi.org/10.1109/TCAD.2013.2284109

[15] R Timothy Marler and Jasbir S Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26, 6 (2004), 369–395.

[16] R. Martins, N. LourenÃğo, S. Rodrigues, J. Guilherme, and N. Horta. 2012. AIDA: Automated analog IC design flow from circuit level to layout. In *2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 29–32. DOI:https://doi.org/10.1109/SMACD.2012.6339409

[17] Carl Edward Rasmussen. 2006. Gaussian processes for machine learning. (2006).

[18] Prakash Kumar Rout and Debiprasad Priyabrata Acharya. 2011. Design of CMOS ring oscillator using CMODE. In *Energy, Automation, and Signal (ICEAS), 2011 International Conference on*. IEEE, 1–6.

[19] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury. 2007. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proc. IEEE* 95, 3 (March 2007), 640–669. DOI:https://doi.org/10.1109/JPROC.2006.889371

[20] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* 104, 1 (Jan 2016), 148–175. DOI:https://doi.org/10.1109/JPROC.2015.2494218

[21] Guido Stehr, Helmut E Graeb, and Kurt J Antreich. 2007. Analog performance space exploration by normal-boundary intersection and by Fourier–Motzkin elimination. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26, 10 (2007), 1733–1748.

[22] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[23] Ronald L Wasserstein and Nicole A Lazar. 2016. The ASA's statement on p-values: context, process, and purpose. (2016).

[24] Z. Yan, P. I. Mak, M. K. Law, and R. P. Martins. 2013. A 0.016-mm^2 144-μW Three-Stage Amplifier Capable of Driving 1-to-15 nF Capacitive Load With >0.95-MHz GBW. *IEEE Journal of Solid-State Circuits* 48, 2 (Feb 2013), 527–540. DOI:https://doi.org/10.1109/JSSC.2012.2229070

[25] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[26] E. Zitzler, K. Deb, and L. Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8, 2 (2000), 173–195.