Batch Bayesian Optimization via Multi-objective Acquisition Ensemble for Automated Analog Circuit Design

Wenlong Lyu [1] Fan Yang [1] Changhao Yan [1] Dian Zhou [1 2] Xuan Zeng [1]

Abstract

Bayesian optimization methods are promising for the optimization of black-box functions that are expensive to evaluate. In this paper, a novel batch Bayesian optimization approach is proposed. The parallelization is realized via a multi-objective ensemble of multiple acquisition functions. In each iteration, the multi-objective optimization of the multiple acquisition functions is performed to search for the Pareto front of the acquisition functions. The batch of inputs are then selected from the Pareto front. The Pareto front represents the best trade-off between the multiple acquisition functions. Such a policy for batch Bayesian optimization can significantly improve the efficiency of optimization. The proposed method is compared with several state-of-the-art batch Bayesian optimization algorithms using analytical benchmark functions and real-world analog integrated circuits. The experimental results show that the proposed method is competitive compared with the state-of-the-art algorithms.

1. Introduction

The advancement of modern society is driven by the development of Integrated Circuits (IC). Unlike the digital circuits where the design flow is already highly automated, the automation of analog circuit design is still a challenging problem.

Traditionally, the design parameters of analog circuits like widths and lengths of transistors are manually calculated by designers with their experience and the understanding of the design specifications. However, due to the progress of IC manufacture technology forecasted by Moore's law, the circuit devices become more and more complicated, and the parasitic effect of the circuits can no longer be ignored. On the other hand, the demands for high-performance, low-power analog circuits are increasing. It is much more difficult to meet the performance and time-to-market requirements with manual circuit design. Automated analog circuit design has thus attracted much research interest in the past decade (Rutenbar et al., 2007).

The analog circuit design automation problems can be formulated as optimization problems. The aim is to find the optimal design parameters that provide the best circuit performance, which can be represented by a figure of merit (FOM) real-valued function. Prior works about analog circuit optimization include offline model-based approaches (Colleran et al., 2003; Daems et al., 2003; Wang et al., 2014) and simulation-based approaches. The offline model-based methods try to build global models of the FOM via manual calculation or regression with simulated data and then optimize the cheap-to-evaluate models. The problem with this approach is that the accurate models are usually hard to get. For example, in Wang et al. (2014), 100,000 randomly simulated points are used to train a sparse polynomial model for an amplifier circuit with ten design parameters.

Simulation-based methods, instead, treat the performances of the circuits as black-box functions. The performances are obtained from circuit simulations. Global optimization algorithms are directly applied to the black-box functions. For simulation-based circuit optimization methods, meta-heuristic algorithms (Phelps et al., 2000; Liu et al., 2009) are widely used. Although these algorithms can explore the whole design space, they have relatively low convergence rate. When the circuit simulation takes a long time, both model-based and simulation-based approaches can be very time-consuming.

In recent years, the Gaussian process (GP) (Rasmussen, 2006) model has been introduced for the automated design of analog circuits to reduce the required number of circuit simulations. In Liu et al. (2014), GP is combined with differential evolution algorithm. Recently, Bayesian optimization (BO) (Shahriari et al., 2016) algorithm has also been applied for analog circuit optimization. In Lyu et al. (2017),

[1] State Key Lab of ASIC and System, School of Microelectronics, Fudan University, Shanghai, China [2] Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, U.S.A. Correspondence to: Fan Yang <yangfan@fudan.edu.cn>, Xuan Zeng <xzeng@fudan.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

Bayesian optimization algorithm is firstly introduced for the single- and multi-objective optimization of general analog circuits and has shown to be much more efficient compared with other simulation-based approaches. In Wang et al. (2017), Bayesian optimization algorithm is combined with adaptive Monte-Carlo sampling to optimize the yield of analog circuits and static random-access memory (SRAM).

Bayesian optimization algorithm is a well-studied algorithm and has demonstrated to be promising for the automated design of analog circuits. However, the standard Bayesian optimization algorithm is sequential. It chooses only one point at each iteration by optimizing the acquisition function. It is often desirable to select a batch of points at each iteration. The sequential property of Bayesian optimization limits its further applications in multi-core computer systems.

Bayesian optimization algorithm has been extended to enable batch selection. Some prior works, like the qEI (Chevalier & Ginsbourger, 2013), qKG (Wu & Frazier, 2016) and parallel predictive entropy search (PPES) (Shah & Ghahramani, 2015) approaches, consider to search for the optimal batch selection for a specific acquisition function. These methods usually involve some approximations or Monte-Carlo sampling, and thus scale poorly as the batch size increases. Other works, including the simulation matching (SM) (Azimi et al., 2010) method, the batch-UCB (BUCB, BLCB for minimization problems) (Desautels et al., 2014) method, the parallel UCB with pure exploration (GP-UCB-PE) (Contal et al., 2013) method, and the local penalization (LP) (González et al., 2016) method, adopted the *greedy* strategies that select individual points until the batch is filled.

All the batch Bayesian optimization algorithms mentioned above choose to use single acquisition function. And except for the SM method (Azimi et al., 2010) and LP method (González et al., 2016) which can use arbitrary acquisition function, other parallelization methods rely on a specific acquisition function. The UCB acquisition function must be used for BUCB and GP-UCB-PE, and the knowledge gradient (KG) acquisition function must be used for the qKG algorithm. As is stated in Hoffman et al. (2011), no single acquisition function can always outperform other acquisition functions. Relying on one acquisition function may result in poor performance.

In this paper, we propose to parallelize Bayesian optimization algorithm via the Multi-objective ACquisition Ensemble (MACE). The proposed MACE method exploits the disagreement between different acquisition functions to enable batch selection. At each iteration, after the GP model is updated, multiple acquisition functions are selected. We then perform multi-objective optimization to find the *Pareto front* (PF) of the acquisition functions. The PF represents the best trade-off between these acquisition functions. When batch evaluations are possible, we can sample multiple points on the PF to accelerate the optimization.

The MACE algorithm is tested using several analytical benchmark functions and two real-world analog circuits, including an operational amplifier with ten design parameters and a class-E power amplifier with twelve design parameters. The BLCB method (Desautels et al., 2014), local penalization method with expected improvement acquisition function (EI-LP) (González et al., 2016), qEI (Chevalier & Ginsbourger, 2013) and qKG (Wu & Frazier, 2016) methods are compared with MACE. The proposed MACE method achieved competitive performance when compared with the state-of-the-art algorithms listed in the paper.

2. Background

In this section, we will present the problem formulation of analog circuit optimization, and review the background of Gaussian process regression and Bayesian optimization.

2.1. Problem Formulation

When designing integrated circuits, the designers have to decide what circuit topology to use and then set the corrsponding design parameters. In this work, we handle the scenarios where the topology of the analog circuit is fixed. This is practical as there are usually a lot of classical topologies for a given design task, so unlike digital circuits, choosing appropriate topology is relatively easy.

Once the circuit topology is fixed, the designer has to choose the appropriate design parameters according to the specifications and the circuit device model. What we want to do is automatically searching for the optimal design parameters. This problem can then be formulated as a bound-constrained black-box optimization problem:

$$\text{minimize } \mathrm{FOM}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} \in \mathrm{D}$ is the vector of design variables, $\mathrm{FOM}(\boldsymbol{x})$ is the objective constructed from the design specifications, the $\mathrm{FOM}(\boldsymbol{x})$ can be deterministic or noisy depending on the design specifications. Given the design parameters $\boldsymbol{x}$, the FOM value can be obtained by commercial circuit simulators like HSPICE or Spectre.

2.2. Gaussian Process Regression

The objective function $\mathrm{FOM}(\boldsymbol{x})$ in (1) can be approximated by Gaussian process (GP) model (Rasmussen, 2006). The GP model is the most commonly used model for Bayesian optimization. The advantage of GP is that it provides a well-calibrated uncertainty of prediction. GP is characterized by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$. In this work, we use squared-exponential ARD kernel (Rasmussen, 2006), and a constant mean function

$m(\boldsymbol{x}) = \mu_0$ for all our experiments. By default, we assume the objective function evaluations are influenced by i.i.d. noise $\epsilon_t \sim N(0, \sigma_n^2)$ and set the noise level σ_n^2 as a hyper-parameter. The introduction of the i.i.d noise also helps to improve the numerical stability.

Denote the training set as $\{X, \boldsymbol{y}\}$ where $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and $\boldsymbol{y} = \{y_1, \dots, y_N\}$, given a new data point $\boldsymbol{x}$, the prediction of $f(\boldsymbol{x})$ is not a scalar value, but a predictive distribution

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

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

$$\begin{array}{rcl} \mu(\boldsymbol{x}) & = & \mu_0 + k(\boldsymbol{x}, X)[K + \sigma_n^2 I]^{-1}(\boldsymbol{y} - \mu_0) \\ \sigma^2(\boldsymbol{x}) & = & k(\boldsymbol{x}, \boldsymbol{x}) - k(\boldsymbol{x}, X)[K + \sigma_n^2 I]^{-1} k(X, \boldsymbol{x}), \end{array} \tag{3}$$

where $k(\boldsymbol{x}, X) = (k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_N))^T$ 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 of the prediction.

2.3. Bayesian Optimization

Bayesian optimization (Shahriari et al., 2016) was proposed for the optimization of expensive black-box functions. It consists of two essential ingredients, i.e., the probabilistic surrogate models and the acquisition functions. The probabilistic surrogate models provide predictions with uncertainties. The acquisition functions make use of the predictive distribution to explore the state space. The procedure of Bayesian optimization is summarized in Algorithm 1.

Algorithm 1 Bayesian Optimization

```
Require: Number of initial sampling points $N_{init}$, number of iterations $N_{iter}$
Randomly sample $N_{init}$ points in the design space
Construct initial GP model
for t = 1, 2, ..., $N_{iter}$ do
    Construct the acquisition function
    Find $\boldsymbol{x}_t$ that optimizes the acquisition function
    Sample $y_t = f(\boldsymbol{x}_t)$
    Update probabilistic surrogate model
end for
Return best $f(\boldsymbol{x})$ recorded during iterations
```

In Bayesian optimization described in Algorithm 1, the acquisition function is used to balance the exploration and exploitation during the optimization. The acquisition function considers both the predictive value and the uncertainty. There are a lot of existing acquisition functions. Examples include the lower confidence bound (LCB), the probability of improvement (PI), and the expected improvement (EI).

The LCB function is defined as follows:

$$\text{LCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \kappa\sigma(\boldsymbol{x}), \tag{4}$$

where the $\mu(\boldsymbol{x})$ and the $\sigma(\boldsymbol{x})$ are the predictive value and uncertainty of GP defined in (3), κ is a parameter that balances the exploitation and exploration.

Following the suggestion of (Srinivas et al., 2010; Brochu et al., 2010), the κ in (4) is defined as:

$$\begin{array}{rcl} \kappa & = & \sqrt{\nu\tau_t} \\ \tau_t & = & 2\log(t^{d/2+2}\pi^2/3\delta), \end{array} \tag{5}$$

where t is the number of current iteration, ν and δ are two user-defined parameters. We fix $\nu = 0.5$ and $\delta = 0.05$ in this paper for the proposed MACE algorithm and our implementation of the BLCB algorithm.

The PI and EI functions are defined as

$$\begin{array}{rcl} \text{PI}(\boldsymbol{x}) & = & \Phi(\lambda) \\ \text{EI}(\boldsymbol{x}) & = & \sigma(\boldsymbol{x})(\lambda\Phi(\lambda) + \phi(\lambda)) \\ \lambda & = & \dfrac{\tau - \xi - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}, \end{array} \tag{6}$$

where τ is the current best value objective value, and ξ is a small positive jitter to improvement the ability of exploration. The $\Phi(.)$ and $\phi(.)$ functions are the CDF and PDF functions of normal distribution. In our implementation of the MACE algorithm, we fix ξ = 1e-3.

There are also other acquisition functions, like the knowledge gradient (Scott et al., 2011) function, predictive entropy search (Hernández-Lobato et al., 2014), and the max-value entropy search(Wang & Jegelka, 2017). A portfolio of several acquisition functions is also possible (Hoffman et al., 2011).

3. Proposed Batch Bayesian Optimization Algorithm

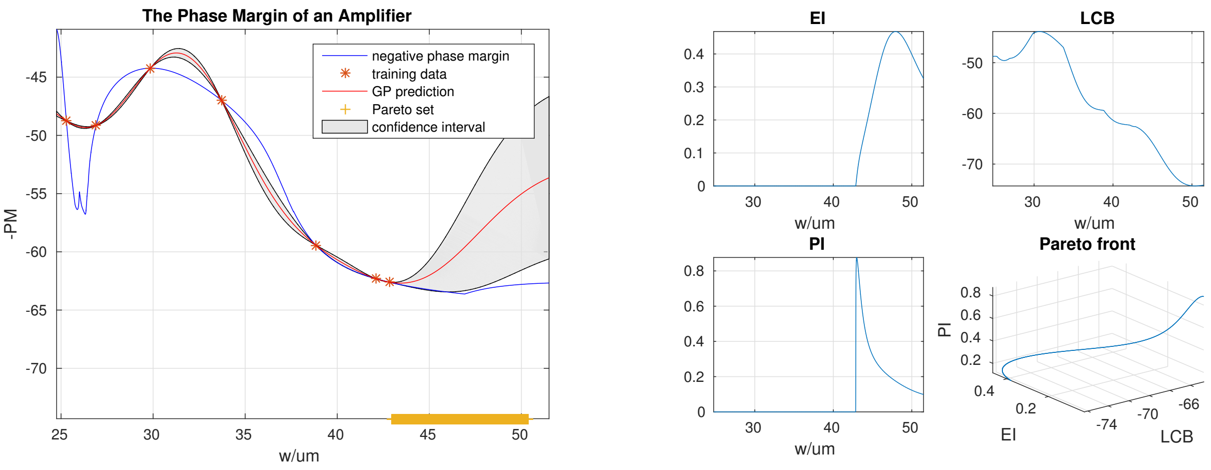
We will present the proposed batch Bayesian optimization algorithm in this section.

3.1. Multi-objective Optimization

Unlike single-objective optimization, there are multiple objectives to optimize in multi-objective optimization problems(Marler & Arora, 2004). The multi-objective optimization problem is formulated as

$$\text{minimize} \quad f_1(\boldsymbol{x}), \ \dots, f_m(\boldsymbol{x}). \tag{7}$$

The multiple objectives to be optimized can be conflicting so that it is usually impossible to find a single solution that is the optimum of all objectives. The goal of multi-objective optimization algorithms is to approximate the *Pareto front* of



(a) True phase margin curve, GP predictions, and Pareto set

(b) The LCB, EI, and PI functions and the Pareto front

Figure 1. Illustration of the multi-objective optimization of acquisition functions

the objectives. A solution $\boldsymbol{x}_1$ is said to *dominate* $\boldsymbol{x}_2$ if $\forall i \in \{1 \dots m\},\ f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ and $\exists j \in \{1 \dots m\}, f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$. A design is *Pareto-optimal* if it is not dominated by any other point in the design space and dominates at least one point. The whole set of the Pareto-optimal points in the design space is called the *Pareto set*, and the set of Pareto-optimal points in the objective space is called the *Pareto front*. It is often unlikely to get the whole Pareto front as there might be infinite points on the Paret front, multi-objective optimization algorithms try to find a set of evenly distributed solutions that approximate the true Pareto front.

There exist many mature multi-objective optimization algorithms, like the non-dominated sorting based genetic algorithm (NSGA-II) (Deb et al., 2002), and the multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang & Li, 2007). In this paper, the multi-objective optimization based on differential evolution (DEMO) (Robič & Filipič, 2005) is used to solve multi-objective optimization problems, but other multi-objective optimization algorithms can also be applied.

3.2. Batch Bayesian Optimization via Multi-objective Acquisition Function Ensemble

Each acquisition function represents a unique selection strategy, different acquisition functions may not agree with each other about where to sample the next point. For example, the value of LCB function always decreases as the $\sigma(\boldsymbol{x})$ increases. However, for the PI function, when $\sigma(\boldsymbol{x})$ increases, the value of PI would decrease when $\mu(\boldsymbol{x}) < \tau$, and increase when $\mu(\boldsymbol{x}) > \tau$. For the EI function, if the function is noiseless, the values of EI function at already sampled points would always be worse than the EI values at any unsampled locations, while this property does not hold for the LCB function.

Algorithm 2 Multi-objective Acquisition Ensemble Algorithm

```
Require: Number of initial sampling points $N_{init}$, number of iterations $N_{iter}$, batch size $B$.
Randomly sample $N_{init}$ points in the design space
Construct initial GP model
for t = 1, 2, ..., $N_{iter}$ do
    Construct the LCB, EI and PI functions according to (4) and (6)
    Find the Pareto front of LCB, EI, PI functions using the DEMO algorithm
    Randomly sample $B$ points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_B$ from the Pareto-optimal points
    Evaluate $\boldsymbol{x}_1, \dots, \boldsymbol{x}_B$ to get $y_1 = f(\boldsymbol{x}_1), \ \dots \ , y_B = f(\boldsymbol{x}_B)$
    Update the GP model
end for
Return best $f(\boldsymbol{x})$ recorded during iterations
```

With multi-objective optimization, the best trade-off between acquisition functions can be captured by the Pareto front of these acquisition functions. We can then sample on the Pareto front to obtain multiple candidate points for the objective function evaluations.

The proposed MACE algorithm is described in Algorithm 2. In the proposed MACE algorithm, the LCB, EI, and PI acquisition functions are selected. Other acquisition functions like KG and PES can also be incorporated into the MACE framework. In each iteration, the following multi-objective

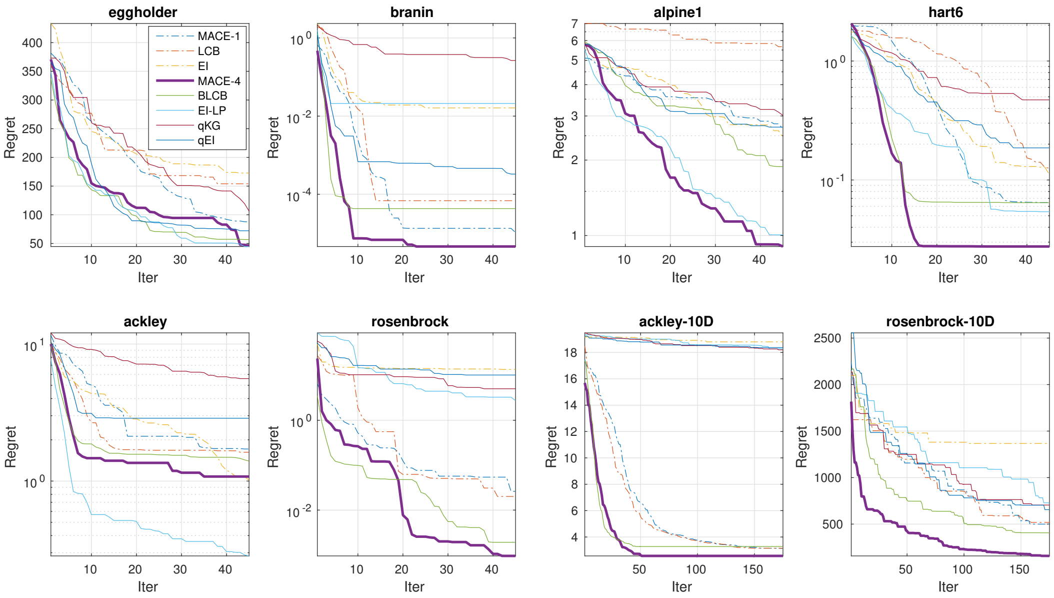


Figure 2. Optimization results of the benchmark functions

optimization problem is constructed:

$$\text{minimize} \quad \text{LCB}(\boldsymbol{x}), \ -\text{EI}(\boldsymbol{x}), \ -\text{PI}(\boldsymbol{x}). \tag{8}$$

Then the DEMO multi-objective optimization algorithm (Robič & Filipič, 2005) is applied to solve the multi-objective problem in (8). Once the Pareto front of LCB, EI and PI is obtained, the candidate points are then randomly sampled from the Pareto front.

In Figure 1, we illustrate the proposed MACE algorithm using an example of a real-world amplifier circuit. The optimization objective is to maximize the phase margin (PM) of the amplifier, so the FOM is defined as $\text{FOM}(\boldsymbol{x}) = -\text{PM}(\boldsymbol{x})$. The width of one of its transistor is the design variable. We sweep the width of the transistor and perform HSPICE simulations to get the FOM values. The curve of FOM values is plotted in Figure 1(a) (the blue line). Several points are randomly sampled from the FOM curve to train the GP model. The LCB, EI, PI functions and the Pareto front of the acquisition functions are plotted in Figure 1(b). We can see from Figure 1(b) that the optimal locations of the three acquisition functions are different, while their best trade-off is captured by the Pareto front. The Pareto set that represents the best trade-off between the three acquisition functions is the interval $[43, 50.4]$, as plotted in Figure 1(a). The candidate points for the next batch of evaluations are randomly sampled from the Pareto set.

4. Experimental Results

The proposed MACE algorithm[1] was tested using eight benchmark functions and two real-world analog circuits. Four state-of-the-art parallel Bayesian optimization methods were compared, including the BLCB algorithm (Desautels et al., 2014), the local penalization method with EI acquisition function (EI-LP) (González et al., 2016), the qEI and qKG methods (Chevalier & Ginsbourger, 2013; Wu & Frazier, 2016). [2]

For the MACE, BLCB, and EI-LP method, the ARD squared-exponential kernel is used and the GP models are fitted by maximum likelihood estimations (MLE); for the qKG and qEI methods, the ARD Matern52 kernels are used, and the GP hyperparameters are integrated via MCMC sampling. The Matern52 kernel and MCMC integration are the default strategies of the qKG and qEI implementations and it is unclear in the documentation about how to change the GP settings.

[1] Available at https://github.com/Alaya-in-Matrix/MACE

[2] We implemented the BLCB algorithm as the available open source implementations only allow discrete input. For the EI-LP method, the code is downloaded from https://github.com/SheffieldML/GPyOpt. The code for qEI and qKG is downloaded from https://github.com/wujian16/Cornell-MOE.

4.1. Benchmark Problems

We tested the MACE algorithm and other parallel BO methods using eight commonly used benchmark functions, as summarized in Table 1.

Table 1. Summary of the analytical benchmark functions

Function	Dimension	Search domain
Branin	2	$[-5, 10] \times [0, 15]$
Alpine1	5	$[-10, 10]^5$
Hartmann6	6	$[0, 1]^6$
Eggholder	2	$[-512, 512]^2$
Ackley2	2	$[-32, 32]^2$
Ackley10	10	$[-32, 32]^{10}$
Rosenbrock2	2	$[-5, 10]^2$
Rosenbrock10	10	$[-20, 20]^{10}$

For all functions except the two 10D functions, we set the number of initial random sampling to $N_{init} = 20$ and the number of iterations to $N_{iter} = 45$. Batch size is set to $B = 4$, the total number of function evaluations is $N_{init} + B \times N_{iter}$. For the 10D Ackley and 10D Rosenbrock functions, we set $N_{init} = 100$ and $N_{iter} = 175$. The experiments were repeated ten times to average the random fluctuations.

We also ran the MACE algorithm in sequential mode and compared with the EI and LCB acquisition functions. The sequential EI and LCB based Bayesian optimization are implemented by setting the batch size $B = 1$ for EI-LP and BLCB respectively.

The mean convergence plots of the tested algorithms on the benchmark functions are given in Figure 2, the statistics of the final regrets are listed in Table 2. As can be seen in Figure 2 and Table 2, when running in sequential mode, the MACE algorithm is competitive with the LCB and EI acquisition functions. The sequential MACE (MACE-1) algorithm gave better performances than the sequential EI (EI-1) and sequential LCB (LCB-1) algorithms in the Eggholder, Branin, Hartmann6, Ackley10, and Rosenbrock10 functions. Also, the parallel MACE (MACE-4) gave the best performances among all the tested algorithms for six out of the eight benchmark functions, and has shown dramatic speedup compared to the sequential MACE. We also performed additional experiments with varied batch sizes, the detail of those experimental results can be seen in the supplementary materials. We report the time spent on the ten-dimensional Rosenbrock function optimization with $B = 4$ as a measure of the algorithm overhead, for the ten-dimensional Rosenbrock function, it took MACE about 11 hours to finish all the $N_{iter} = 175$ iterations, the BLCB algorithm took about five hours, for the EI-LP algorithm, it took only one hour to finish the optimization. The overheads for qEI and qKG are much larger, it took more than two days for qKG and qEI to finish the optimization of the ten-dimensional Rosenbrock function.

4.2. Operational Amplifier

The operational amplifier (Wang et al., 2014) shown in Figure 3 is used to test Bayesian optimization algorithms. The circuit is designed using the 180nm process. It has 10 design parameters, including the lengths and widths of transistors, the resistance of the resistors and the capacitance of the capacitors. The circuit is simulated using the commercial HSPICE circuit simulator.

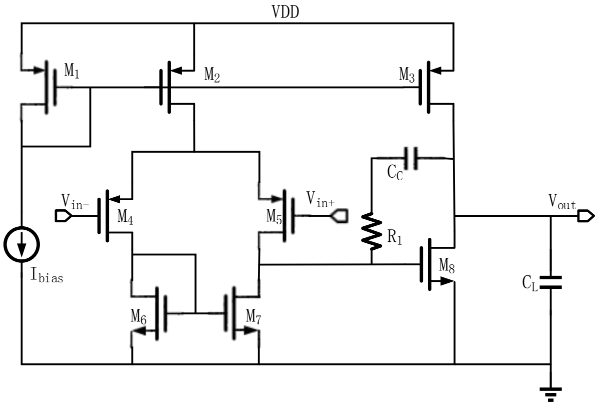


Figure 3. Schematic of the operational amplifier

We want to maximize the gain, unit gain frequency (UGF) and the phase margin (PM) for this amplifier. The Figure of Merit FOM is constructed as

$$FOM = -1.2 \times gain - 10 \times UGF - 1.6 \times PM.$$

For this circuit, we compared the MACE algorithm with the BLCB and EI-LP algorithms. The qKG and qEI are not compared as the computation of qEI and qKG acquisition functions become very slow for the ten-dimensional functions.

We run the algorithms in sequential mode and batch mode. For the batch mode, the batch size is set to $B = 4$. The number of initial random sampling is set to $N_{init} = 100$, and the number of iterations is set to $N_{iter} = 100$.

The mean convergence plot for the sequential and batch runs are given in Figure 4. The mean and standard deviation of the final optimized FOM values are listed in Table 3. As can be seen, on average, the batch MACE algorithm had the fastest convergence rate compared with the sequential MACE algorithm and other parallel algorithms. It should also be noted that the final optimized FOM values given by MACE-4 have very small deviation (0.105) compared with other algorithms.

Table 2. Statistics of the regrets of the benchmark functions

	Eggholder	Branin	Alpine1	Hartmann6
MACE-1	87.65±75.83	1.05e-5±1.31e-5	2.66305±1.05844	0.0646869±0.0621189
LCB-1	153.9±112.8	6.86e-5±1.13e-4	5.66812±1.76973	0.125565±0.122684
EI-1	172.8±132.2	1.62e-2±1.63e-2	2.46061±1.56079	0.110561±0.146809
MACE-4	46.38±40.89	**4.62e-6±6.64e-6**	**0.903805±0.835209**	**0.0275738±0.052254**
BLCB-4	56.86±35.91	4.32e-5±6.33e-5	1.8843±0.938873	0.06447±0.0621176
EI-LP-4	**44.68±56.45**	2.11e-2±1.84e-2	1.0059±0.456865	0.0540446±0.0558557
qKG-4	106.4±67.64	2.65e-1±2.70e-1	3.01513±1.13414	0.47134±0.18939
qEI-4	72.13±52.08	3.29e-4±1.14e-3	2.7074±1.05145	0.186088±0.116323
	Ackley2	Rosenbrock2	Ackley10	Rosenbrock10
MACE-1	1.71474±1.12154	0.026173±0.051189	3.1348±0.447874	499.697±300.899
LCB-1	1.624±0.926437	0.0201124±0.0205367	3.14797±0.519164	517.944±288.955
EI-1	1.0136±0.985858	13.5508±9.52734	18.8006±0.652136	1367.08±637.507
MACE-4	1.07906±0.886466	**0.00095416±0.00093729**	**2.56439±0.535488**	**158.116±50.0024**
BLCB-4	1.40051±1.02849	0.00191986±0.00180895	3.27543±0.735501	406.819±127.351
EI-LP-4	**0.284265±0.24634**	2.73645±2.05923	18.2682±0.608564	721.351±327.365
qKG-4	5.59394±1.80595	5.03976±3.72014	18.197±0.764103	705.112±412.762
qEI-4	2.87373±1.02405	10.1881±15.0432	18.3686±0.501869	655.208±340.954

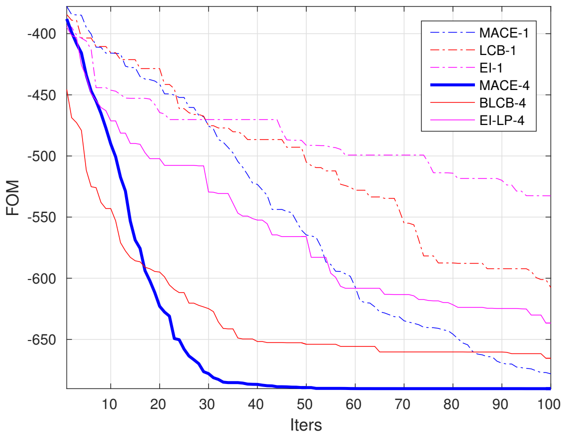


Figure 4. Optimization results of the operational amplifier

Table 3. Optimization results of the operational amplifier

Algorithm	Results
MACE-1	-678.174±21.7445
LCB-1	-607.583±51.9786
EI-1	-532.555±66.942
MACE-4	**-690.3±0.104963**
BLCB-4	-665.442±23.066
EI-LP-4	-636.675±35.7359

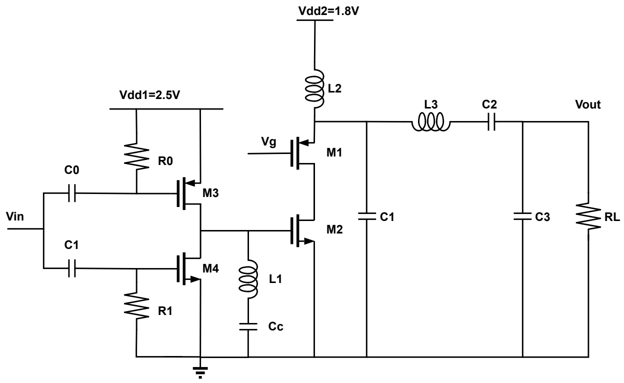


Figure 5. Schematic of the power amplifier

4.3. Class-E Power Amplifier

The class-E power amplifier shown in Figure 5 is used to test Bayesian optimization algorithms. The circuit is designed using the 180nm process with 12 design parameters, the circuit is simulated by the commercial HSPICE circuit simulator to get its performances.

For this power amplifier, we aim to maximize the power added efficiency (PAE) and the output power (Pout), the Figure of Merit FOM is constructed as

$$FOM = -3 \times PAE - Pout.$$

The MACE, BLCB, and EI-LP algorithms were tested in both sequential and batch modes. The number of initial sampling is $N_{init} = 100$. The number of iterations is

$N_{iter} = 100$. The batch size is set to $B = 4$. The total number of HSPICE simulations is 500 for each batch run and 200 for each sequential run.

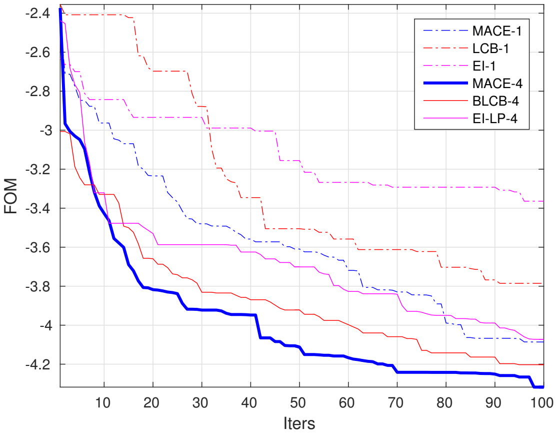


Figure 6. Optimization results of the class-E power amplifier

The optimization results of the class-E power amplifier are given in Figure 6 and Table 4. We can see that the MACE outperformed the BLCB and EI-LP in both sequential and batch mode. For the batch runs, the MACE converges fastest among the three algorithms, while the sequential MACE (MACE-1) has comparable performance as the batch EI-LP (EI-LP-4) method.

Table 4. Optimization results of the power amplifier

Algorithm	Results
MACE-1	-4.08608±0.296647
LCB-1	-3.78533±0.335532
EI-1	-3.36407±0.307489
MACE-4	**-4.31762±0.347026**
BLCB-4	-4.20266±0.211102
EI-LP-4	-4.07233±0.244436

5. Conclusion

In this paper, a batch Bayesian optimization algorithm is proposed for the automation of analog circuit design. The parallelization is achieved via the multi-objective ensemble of acquisition functions. In each iteration, the candidate points are sampled from the Pareto front of multiple acquisition functions. We compared the proposed MACE algorithm using analytical benchmark functions and real-world circuits, it is shown that the MACE algorithm is competitive compared with the state-of-the-art methods listed in the paper.

Acknowledgements

This research was supported partly by the National Major Science and Technology Special Project of China (2017ZX01028101-003), partly by National Key Research and Development Program of China 2016YFB0201304, 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

Azimi, J., Fern, A., and Fern, X. Z. Batch bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems*, pp. 109–117, 2010.

Brochu, E., Cora, V. M., and De Freitas, N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Chevalier, C. and Ginsbourger, D. Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, pp. 59–69. Springer, 2013.

Colleran, D. M., Portmann, C., Hassibi, A., Crusius, C., Mohan, S. S., Boyd, S., Lee, T. H., and del Mar Hershenson, M. Optimization of phase-locked loop circuits via geometric programming. In *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003.*, pp. 377–380, Sept 2003. doi: 10.1109/CICC.2003.1249422.

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–240. Springer, 2013.

Daems, W., Gielen, G., and Sansen, W. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(5):517–534, May 2003. ISSN 0278-0070. doi: 10.1109/TCAD.2003.810742.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

González, J., Dai, Z., Hennig, P., and Lawrence, N. Batch bayesian optimization via local penalization. In *Artificial Intelligence and Statistics*, pp. 648–657, 2016.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pp. 918–926, 2014.

Hoffman, M. D., Brochu, E., and de Freitas, N. Portfolio Allocation for Bayesian Optimization. In *UAI*, pp. 327–336, 2011.

Liu, B., Wang, Y., Yu, Z., Liu, L., Li, M., Wang, Z., Lu, J., and Fernández, F. V. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration, the VLSI journal*, 42(2):137–148, 2009.

Liu, B., Zhao, D., Reynaert, P., and Gielen, G. G. E. 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):169–182, Feb 2014. ISSN 0278-0070. doi: 10.1109/TCAD.2013.2284109.

Lyu, W., Xue, P., Yang, F., Yan, C., Hong, Z., Zeng, X., and Zhou, D. An efficient bayesian optimization approach for automated optimization of analog circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2017.

Marler, R. T. and Arora, J. S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

Phelps, R., Krasnicki, M., Rutenbar, R. A., Carley, L. R., and Hellums, J. R. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(6):703–717, Jun 2000. ISSN 0278-0070. doi: 10.1109/43.848091.

Rasmussen, C. E. Gaussian processes for machine learning. 2006.

Robič, T. and Filipič, B. Demo: Differential evolution for multiobjective optimization. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E. (eds.), *Evolutionary Multi-Criterion Optimization*, pp. 520–533, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31880-4.

Rutenbar, R. A., Gielen, G. G. E., and Roychowdhury, J. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proceedings of the IEEE*, 95(3):640–669, March 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.889371.

Scott, W., Frazier, P., and Powell, W. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

Shah, A. and Ghahramani, Z. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, pp. 3330–3338, 2015.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2494218.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 1015–1022. Omnipress, 2010.

Wang, M., Lv, W., Yang, F., Yan, C., Cai, W., Zhou, D., and Zeng, X. Efficient yield optimization for analog and sram circuits via gaussian process regression and adaptive yield estimation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

Wang, Y., Orshansky, M., and Caramanis, C. Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2014.

Wang, Z. and Jegelka, S. Max-value entropy search for efficient bayesian optimization. *arXiv preprint arXiv:1703.01968*, 2017.

Wu, J. and Frazier, P. The parallel knowledge gradient method for batch bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 3126–3134, 2016.

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