LoCoMOBO: A Local Constrained Multiobjective Bayesian Optimization for Analog Circuit Sizing

Konstantinos Touloupas, *Student Member, IEEE*, and Paul P. Sotiriadis, *Senior Member, IEEE*

***Abstract*—A local constrained multiobjective Bayesian optimization (LoCoMOBO) method is introduced to address automatic sizing and tradeoff exploration for analog and RF integrated circuits (ICs). LoCoMOBO applies to constrained optimization problems utilizing multiple Gaussian process (GP) models that approximate the objective and constraint functions locally in the search space. It searches for potential pareto optimal solutions within trust regions of the search space using only a few time-consuming simulations. The trust regions are adaptively updated during the optimization process based on feasibility and hypervolume metrics. In contrast to mainstream Bayesian optimization approaches, LoCoMOBO uses a new acquisition function that can provide multiple query points, therefore allowing for parallel execution of costly simulations. GP inference is also enhanced by using GPU acceleration in order to handle highly constrained problems that require large sample budgets. Combined with a framework for schematic parametrization and simulator calls, LoCoMOBO provides improved performance tradeoffs and sizing results on three real-world circuit examples, while reducing the total runtime up to ×43 times compared to state-of-the-art methods.**

***Index Terms*—Analog sizing, Bayesian methods, machine learning, optimization.**

I. Introduction

THE CONTINUOUS advancements in CMOS-integrated circuit (IC) technology have resulted in complex circuits with increased functionalities and low area utilization. Although digital IC circuits push the boundaries of electronic systems in state-of-the-art applications, analog and RF ones remain indispensable [1]–[3]. Applications such as Internet of Things that impose new levels of integration for sensor interfaces and communication systems are shifting the design effort toward analog IC design, with new architectures and novel computing paradigms being put forward [4]–[6]. However, manual analog design is becoming increasingly cumbersome, as transistor scaling and the demands for rapid prototyping lead to either prolonged design cycles or suboptimal designs. This bottleneck in system design [7], caused by the difficulty to facilitate rapid analog IC design, is addressed by adopting electronic design automation (EDA) methodologies.

Methodologies for automated design of analog circuits include optimization methods or topology-specific heuristics. In the general case of optimization-based sizing, given a fixed circuit topology and a set of associated parameters, a computer software determines the optimal parameter values that satisfy the designer's set of specifications. This is addressed by using either closed-form equations [8]–[11] or computer simulations for evaluation [12]–[18]. In this work, we focus on simulation-based optimization.

Black-box population-based algorithms have been extensively studied in the context of automated analog sizing. These include particle swarm optimization (PSO) [14], genetic algorithms (GAs) [17], and simulated annealing (SA) [19] and provide with estimates for global optima by using many evaluations. Their main weakness is the treatment of expensive optimization problems, where the objective and constraint function evaluations are time consuming. Analog automated sizing is one such problem, since simulation for large-scale circuits may take hours to complete. This problem is only deteriorated in multiobjective optimization problems (MOPs), where multiple specification requirements drive the circuit sizing. To address this issue, surrogate-assisted optimization methods have been proposed, which use fast to evaluate surrogate models to reduce the number of costly evaluations.

In surrogate-assisted methods, past evaluations are modeled and used to guide the optimization search, either by replacing expensive function evaluations or by prescreening candidate query points for evaluation [20]. Popular techniques include support vector machines [21], artificial neural networks [12], sparse regression methods [13], and Gaussian processes [20]. An important aspect of this approach is the quality of the objective function approximations. Depending on the problem difficulty, the employed models need large amounts of data (acquired by simulations), which is in contradiction to the small-data nature of the problem. Local-based surrogate methods have been put forward recently [12], providing a better alternative to global surrogate models, since they are easier to scale in high-dimensional problems.

Among the class of surrogate-assisted optimization methods, Bayesian optimization (BO) is the most promising one, being studied across many disciplines, such as machine learning [22], chemical design [23], and robotics [24] . BO has been applied to automated analog and RF circuit sizing [25]–[28] with promising results. BO methods, in contrast to other surrogate based ones, do not use the model to approximate the objective function(s) over the entire search space, but make use of the uncertainty information from Gaussian process (GP) models to directly guide the search toward the global optimum.

BO relies on two factors to provide with the global minimum, a GP surrogate model, and an acquisition function

Manuscript received 14 April 2021; revised 22 July 2021; accepted 23 September 2021. Date of publication 19 October 2021; date of current version 22 August 2022. This work was supported by Greece and the European Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" under Grant MIS-5000432, implemented by the State Scholarships Foundation (IKY). This article was recommended by Associate Editor P. Li. *(Corresponding author: Konstantinos Touloupas.)*

The authors are with the Department of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Greece (e-mail: ktouloupas@mail.ntua.gr).

Digital Object Identifier 10.1109/TCAD.2021.3121263

that defines a score of utility for evaluation of each point on the search space. By optimizing the fast-to-evaluate acquisition function, a query point for objective function evaluation is determined. This makes BO the state-of-the-art approach to low-budget optimization problems. However, an important limitation stems from its difficulty to handle high-dimensional problems and large numbers of samples. In particular, GP models rely on the Euclidean distance to define sample correlations and become impractical in high dimensions [29]. Moreover, the computational complexity for GP inference becomes prohibitive when many simulation data need to be used [30]. To leverage the benefits of BO in large-scale problems, extensive studies were proposed, but they are limited to single objective or unconstrained settings.

As analog and RF circuit design reduces to exploration of competing specifications, an efficient automated sizing framework should be able to determine optimal tradeoffs while satisfying constraints. This tradeoff exploration [31] process provides insights for the selection of device sizes and can be addressed using multiobjective optimizers. However, methods that address constrained MOPs in a sample-efficient manner have not been thoroughly studied in the context of analog ICs. In [28], a multiobjective variant of BO, WEIBO, was proposed to handle constrained problems, but it inherited the problems of classical BO in terms of the curse of dimensionality. In addition, in [32], an entropy-based BO, MESMOC, was introduced to handle constrained MOPs. Both of the aforementioned algorithms are sequential in nature, which restricts them from fully utilizing modern hardware systems to parallelize the optimization process. In [33], an additional multiobjective BO variant was introduced for analog circuit sizing, but it is limited to unconstrained MOPs.

Motivated by the above, a new multiobjective BO algorithm that handles constraints, local constrained multiobjective BO (LoCoMOBO), is proposed in this work, to address the analog and RF IC sizing problem in a sample-efficient manner. LoCoMOBO utilizes a local-based approach that maintains separate GP models in promising subregions of the search space, therefore enhancing BO efficiency. This boosts the predictive capabilities of GP models in small regions of the search space and reduces extensive exploration. In contrast to classical BO and other variants, LoCoMOBO is able to provide with multiple query points; therefore, it enables the use of parallel simulations to speed up the optimization process. A modified Thompson sampling (TS) [34] acquisition function is used, which ranks query points based on their constraint violation degree and their contributing Hypervolume. To address the challenge of computational complexity regarding training times for GP models, GPU acceleration and a batch method to simultaneously train multiple GP models are also used within LoCoMOBO. The experimental results using benchmark functions and three real-world circuits suggest that LoCoMOBO provides with better tradeoff information and reduces the total runtime of the optimization compared to state-of-the-art approaches.

The remainder of this article is structured as follows. Section II provides a background on analog and RF IC sizing and a necessary introduction to GP modeling and BO. Section III presents LoCoMOBO and provides empirical evidence of its benefits on a set of benchmark functions. Section IV provides three experimental optimization results, on two CMOS amplifiers and a low noise amplifier (LNA) circuit. Finally, Section V concludes this article and summarizes the results.

II. Background and Problem Formulation

To elucidate the benefits of LoCoMOBO, we briefly introduce the circuit optimization formulation used and discuss the concepts of GP Regression and BO.

A. Problem Formulation

Analog and RF circuit sizing can be cast to a constrained optimization problem

$$\begin{aligned} \min\ & F(\mathbf{x}), \mathbf{x} = [x_1, x_2, \ldots, x_d] \\ \text{s.t.}\ & g_j(\mathbf{x}) \leq 0, j = 1, \ldots, l \\ & L_i \leq x_i \leq U_i, i = 1, \ldots, d \end{aligned} \tag{1}$$

where vector $\mathbf{x}$ contains the design variables, L_i and U_i are the lower and upper bounds of the ith variable, $\mathbb{S} = \prod_{i=1}^{d} [L_i, U_i]$ is the variable space, F is the objective (fitness) function, and g_j is the jth constraint function. For a given parameter vector $\mathbf{x}$, its degree of constraint violation is defined as

$$\mathrm{CV}(\mathbf{x}) = \sum_j \max\big[0, g_j(\mathbf{x})\big]. \tag{2}$$

Equality constraints can also be handled after being relaxed to inequalities, given simple transformations [35].

In the case of MOPs, F is a vector of m (often conflicting) functions $F(\mathbf{x}) = [f_i(\mathbf{x})]_{i=1}^{m}$. Minimizing F typically results in multiple solutions that constitute a Pareto set (PS). For a point $\mathbf{x}^\star$ to be a pareto optimum, i.e., to belong in the PS, every variation of $\mathbf{x}^\star$ further minimizing one of the m objective functions $f_i(\mathbf{x}^\star)$ must also deteriorate another one. We say that a feasible solution $\mathbf{x}_1$ dominates another feasible solution $\mathbf{x}_2$, if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ for $i = 1, \ldots, m$ and there exists $m' \in [1, \ldots, m]$ such that $f_{m'}(\mathbf{x}_1) < f_{m'}(\mathbf{x}_2)$. Consequently, $\mathbf{x}_1$ belongs to the PS if it is feasible and it is not pareto dominated by any other $\mathbf{x} \in \mathbb{S}$. We also use the term Pareto front (PF) to denote the image of PS via function F, i.e., $\mathrm{PF} = \{F(\mathbf{x}) | \mathbf{x} \in \mathrm{PS}\}$.

In an arbitrary finite subset $P = \{\mathbf{x}_i\}_{i=1}^{N}$ of $\mathbb{S}$, multiple levels of pareto dominance (subsets of P) may be defined. The first level is the subset of P including all vectors that are nondominated by any other vector in P, and only those. The first level is an approximation of PS of the MOP. The second level is defined by removing the first level (approximate PS) from P, and keeping the nondominated solutions from this set. The process is repeated until all samples in P are assigned to a dominance level.

In the context of analog circuit sizing, a PF provides designers with a model of the conflicting relationships of circuit performance aspects, aiding the device sizing. To utilize optimization algorithms in practice, dedicated software frameworks have been proposed [36], [37] automating the process of simulator calls, circuit netlist parametrization, results parsing, and performance computation.

In this work, we use an in-house interface, written in Python, connecting the Cadence Spectre Simulator with our in-house high-level optimization tool by computing user-defined metrics and by processing the simulator's outputs. In addition, our interface allows us to run multiple simulations using both the simulator's batch mode and parallel processing.

B. Gaussian Processes

Consider a variable space $\mathbb{S}$ and let $\mathbf{x}_i$, $i = 1, 2, \ldots, N$ be the inputs in $\mathbb{S}$ of an unknown function $f : \mathbb{S} \to \mathbb{R}$. The

observations are $y_i = f(\mathbf{x}_i) + \epsilon_i$, $i = 1, \ldots, N$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ is uncorrelated additive noise. The inputs $\{\mathbf{x}_i\}$ and observations $\{y_i\}$ are arranged in pairs forming a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.

A GP is a stochastic process of infinitely many random variables that define a probability distribution over functions on $\mathbb{S}$. GPs are defined by a mean function $m : \mathbb{S} \to \mathbb{R}$ and a kernel function $k : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$. By assuming that f follows a GP:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \tag{3}$$

one defines a probabilistic surrogate of f to use for prediction. GPs have the following defining property. For every positive integer N, vector $\mathbf{f} = [f(\mathbf{x}_i)]_{i=1}^N$ follows a Multivariate Gaussian distribution:

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, K). \tag{4}$$

Here, the $(N \times 1)$ mean vector is $\boldsymbol{\mu} = [m(\mathbf{x}_i)]_{i=1}^N$ and the $(N \times N)$ covariance matrix K is defined such that $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij}$, where δ_{ij} is the Kronecker delta.

When no prior information about $f(\mathbf{x})$ is available, $m(\mathbf{x})$ is set to zero. The kernel function defines a measure of similarity between the outputs of any two input points. Popular kernels include the RBF and Matèrn families [30]. In this work, we use the Matèrn 5/2 kernel that provides with twice differentiable functions [38]

$$k\big(\mathbf{x}_i, \mathbf{x}_j\big) = \sigma^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) e^{-\sqrt{5}r} \tag{5}$$

where

$$r = \left(\sum_{k=1}^{d} \frac{(x_{i,k} - x_{j,k})^2}{\lambda_k^2}\right)^{1/2}. \tag{6}$$

Lengthscales λ_k and parameters σ_n and σ are called the hyperparemeters of the GP model and are denoted by $\boldsymbol{\theta}$. Learning $\boldsymbol{\theta}$ from data adjusts the GP model to the observations in $\mathcal{D}$. This is done using gradient descent optimization to minimize the negative log marginal likelihood

$$L(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} + \frac{1}{2}\log(|K|) + \frac{N}{2}\log(2\pi) \tag{7}$$

where $\mathbf{y}^T$ is a $(1 \times N)$ vector of the observations.

To predict $f(\mathbf{x})$ at a point $\mathbf{x}^\star \notin \mathcal{D}$, one uses the predictive distribution $p(f(\mathbf{x}^\star)|\mathcal{D})$. This is a Gaussian distribution with mean and variance

$$\begin{aligned} \mu_{f|D}\big(\mathbf{x}^\star\big) &= \mathbf{k}^{\mathrm{T}} K^{-1}\mathbf{y} \\ \sigma^2_{f|D}\big(\mathbf{x}^\star\big) &= c - \mathbf{k}^{\mathrm{T}} K^{-1}\mathbf{k}. \end{aligned} \tag{8}$$

Here, $\mathbf{k}^{\mathrm{T}}$ is a $(1 \times N)$ vector with values $k(\mathbf{x}_i, \mathbf{x}^\star)$ for $i = 1, \ldots, N$ and $c = k(\mathbf{x}^\star, \mathbf{x}^\star)$. The predictive distribution provides estimates not only for pointwise query points but for multiple inputs as well. In this case, equations in (8) provide the mean and the diagonal elements of the covariance matrix of a multivariate Gaussian distribution $p([f(\mathbf{x}_1^\star), \ldots, f(\mathbf{x}_k^\star)]|\mathcal{D})$. The covariance between two points $\mathbf{x}^\star, \mathbf{x}^{\star\prime}$ is given by [30]

$$\mathrm{Cov}\big(\mathbf{x}^\star, \mathbf{x}^{\star\prime}\big) = k\big(\mathbf{x}^\star, \mathbf{x}^{\star\prime}\big) - \mathbf{k}^{\mathrm{T}}_{X,\mathbf{x}^\star} K^{-1} \mathbf{k}_{X,\mathbf{x}^{\star\prime}} \tag{9}$$

where the $(1 \times N)$ vector $\mathbf{k}^{\mathrm{T}}_{X,\mathbf{x}^\star} = [k(\mathbf{x}_i, \mathbf{x}^\star)]_{i=1}^N$.

In terms of computational complexity, GP training and query point prediction require $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$ time, respectively. In addition, drawing N_S samples over k query points requires $\mathcal{O}(kN^2 + k^2(N + N_S) + k^3)$ time [39].

C. Bayesian Optimization

BO [34] is a low-budget method to solve global optimization problems involving expensive-to-evaluate black-box functions. Considering a real valued cost function f, BO learns a fast-to-evaluate probabilistic surrogate model of f from its past evaluations. Starting from an initial set of evaluations, each query point is selected sequentially, by balancing exploration and exploitation to find the global optimum.

The BO framework consists of two main components; the GP model that approximates function f and an acquisition function. Popular acquisition functions, such as expected improvement (EI), probability of improvement (PI), and lower confidence bound (LCB) [34], make use of pointwise GP predictive distributions to assign a score of utility for expensive evaluation to each point in the search space. To determine the best candidate point, off-the-shelf optimization algorithms such as DIRECT or CMA-ES [34] are used to maximize the fast-to-evaluate acquisition function.

In the case of MOPs, a popular BO approach is the reduction of the multiobjective problem to a single-objective one, using random objective function scalarizations [28]. When constraints apply, these approaches are extended by using separate GP models to approximate the black-box constraint functions. Using pointwise predictive distributions of the constraint models, each point in the variable space is assigned a probability of feasibility, which is used to weight acquisition functions and eventually drive the search to feasible solutions [28]. In the aforementioned category falls the WEIBO algorithm, which is compared against the proposed method in the following sections. Another algorithm, MESMOC [32], uses a Monte-Carlo-based entropy approximation to select a single query point by utilizing the output space entropy to maximize the information gain about the optimal PF.

III. Proposed Approach

Expanding BO to multiple objectives and constrained high-dimensional problems are two critical challenges for an efficient low-budget MOP optimizer. In high-dimensional search spaces, GP models result in high predictive uncertainties, thereby encouraging exploration on top of exploitation of promising subregions of the design space [40].

In addition, query point selection is not straightforward in the multiobjective setting, since it must take into account pareto dominance and diversity. To address the aforementioned issues, we need to resort to new model-based approaches that exploit better the promising regions and define corresponding acquisition functions.

This section describes the local-based LoCoMOBO algorithm. First, a local-search scheme used to circumvent the problem of GP inefficiency in high-dimensional problems is presented. Then, a multiobjective acquisition function is proposed that is able to drive the BO to feasible and pareto optimal solutions.

A. BO Local-Based Approach

Starting from an initial estimate $\mathbf{x}_c$ of the point of global optimum (or in this case, a PS), searching locally for an improvement is the purpose of local-based optimization methods. Typically, in such cases, an approximation $\hat{F}$ of the objective function F is used to determine future query points according to

$$\mathbf{x}^\star = \operatorname{argmin} \hat{F}(\mathbf{x}), \quad ||\mathbf{x} - \mathbf{x}_c|| \leq R. \tag{10}$$

Here, $\mathbf{x}_c$ is the current best solution and R is a trust radius, restricting the next query point to be near the current best. If the expensive evaluation of $F(\mathbf{x}^\star)$ leads to a better solution, the current best is updated. In our case, $\hat{F}$ relates indirectly to the actual problem defined in (1), since it provides a metric upon the multiobjective optimization progress (see Section III-B).

Based on the aforementioned general scheme for local-based optimization, a similar single objective BO was proposed in [40], where query point selection is restricted within a hypercube centered at $\mathbf{x}_c$, with its edge denoted by $\mathcal{L}$. GP models are trained using a dataset (archive) consisting of past objective function evaluations. By setting the maximum utility point in the archive as $\mathbf{x}_c$, this approach restricts excessive exploration of the search space. This approach is extended here to address constrained MOPs. By using $N_{TR} \geq 1$ trust regions in parallel, each one with their separate center and past archive, multiple paths toward the optimal subregions of the variable space are followed and global search is enhanced. This is particularly helpful in the case of multimodal functions.

To extend the trust region approach to the constrained multiobjective case, one should define a measure of the *utility of the evaluated points*. This is imperative in order to select the trust region center $\mathbf{x}_c$, but it is not straightforward, since typically multiple pareto optimal solutions exist. Starting from the requirement for well spread pareto optimal solutions in the objective space, the trust region center is selected as follows. In each iteration, every trust region has a local PF and PS, namely, PF_i and PS_i, which are determined among the samples from their respective archive $\mathcal{D}_i$ that reside in the trust region's hypercube. The global archive $\mathcal{D} = \bigcup_i [\mathcal{D}_i]$ is used to determine the global PF and PS. The selection of the ith trust region center is done based on the following rule.

1) If no feasible sample exists in $\mathcal{D}_i$, $\mathbf{x}_{c,i}$ is the sample in $\mathcal{D}_i$ with the minimum constraint violation (2).
2) If feasible samples exist in $\mathcal{D}_i$ and $A = \text{PS}_i \cap \text{PS} \neq \emptyset$, $\mathbf{x}_{c,i}$ is the the maximum crowding distance sample in A.
3) If feasible samples exist in $\mathcal{D}_i$ and $A = \text{PS}_i \cap \text{PS} = \emptyset$, $\mathbf{x}_{c,i}$ is the sample from PS_i being in the topmost global dominance level and having the maximum crowding distance.

The crowding distance metric [41] quantifies the diversity of each solution in the objective space. It is used to compare solutions that are on the same dominance level. Since the crowding distance of the samples that lie on the edges of the PF is infinite, we use the maximum finite values to determine the trust region centers.

The optimizer operates on normalized search space $\mathbb{S}_n = [0, 1]^d$, and query points are transformed back to the original search space prior to evaluation. This facilitates GP training and helps define trust region lengths $\mathcal{L}_i$ in the normalized space, irrespective of the variable domain boundaries. For the ith trust region, its hypercube occupies a space $\mathbb{S}_n^{(i)} = \{\mathbf{x} | \mathbf{x}_{L,i} \leq \mathbf{x} \leq \mathbf{x}_{U,i}\}$ with $\mathbf{x}_{L,i} = \max(0, \mathbf{x}_{c,i} - \mathcal{L}_i)$ and $\mathbf{x}_{U,i} = \min(1, \mathbf{x}_{c,i} + \mathcal{L}_i)$.

Each space $\mathbb{S}_n^{(i)}$ is updated according to the rate with which the optimizer finds new maximum utility points $\mathbf{x}_{c,i}$. In particular, a simple method is used that counts the successes and failures of the optimizer to find better solutions. After a user-specified number of successes (failures) for the ith trust region, $\mathcal{L}_i$ is increased (decreased) by a user-specified factor $\rho > 1$, i.e.,

$$\mathcal{L}_i := \begin{cases} \rho \cdot \mathcal{L}_i, & \text{for consecutive successes} \\ 1/\rho \cdot \mathcal{L}_i, & \text{for consecutive failures.} \end{cases} \tag{11}$$

Therefore, in cases where the surrogate model's predictions lead to improvements upon the optimization goal, the trust region edge increases; otherwise, it shrinks to restrict the search closer to the current best point. For the rest of this work, ρ is considered constant and equal to 1.2 to allow for smooth transition between trust-region sizes.

Fig. 1 shows an example of the local-based approach, using two trust regions and a function $F(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^2$, over a bounded space $[-2, 2]^2$. In this example, no constraints apply. After an initial sampling of the variable space, the samples are clustered to two trust regions, using k-means clustering. In each trust region [shown in red-dotted boxes in Fig. 1(a) and (b)], a new solution (marked as a diamond) is selected and evaluated. The trust region centers are chosen so as to facilitate the diversification of the global PF solutions, shown in the objective space plot in Fig. 1(c). Once the query points are evaluated, the trust regions move to new locations shown in light red-dotted boxes.

B. Acquisition Function

To address the constrained multiobjective selection, LoCoMOBO uses a composite selection scheme where both feasibility criteria and the pareto optimality are taken into account. The proposed acquisition function is based on TS [34]. TS is a randomized selection strategy addressing the exploration–exploitation tradeoff by drawing random samples from the posterior distribution of the GP models and selecting a query vector by optimizing on these samples. Recent works highlight the suitability of TS acquisition functions for parallelizing BO [42]. By drawing $N_S > 1$ samples from the GP posteriors, and finding a single maximum utility point for each one of them, one can have multiple query points to evaluate in parallel [42]. This is useful in cases where time-consuming function evaluations can be run in parallel, such as in our case, since parallel function evaluations provide with more information in the same time frame.

Sampling functions from GP models, however, would require computing the joint predictive distribution over infinitely many points, which is not possible. A common practice is to find the maximizer of a discretization over the search domain [40]. Although this is a simple approach and works in many cases, it is not suitable for high-dimensional problems, where exponentially many discrete points need to be used to achieve a good coverage. To circumvent this limitation, we sample GP models by building an analytic approximation to the samples, using random fourier features (RFFs) [43], [44].

A GP posterior sample $\tilde{f}(\mathbf{x})$ can be approximated by a linear model $\tilde{f}(\mathbf{x}) \approx \phi(\mathbf{x})^T \boldsymbol{\theta}$, where $\phi : \mathbb{S} \to \mathbb{R}^M$ is a finite-dimensional basis function that approximates the GP kernel as $k(\mathbf{x}, \mathbf{x}') \approx \phi(\mathbf{x})^T \phi(\mathbf{x}')$. Vector $\boldsymbol{\theta} \in \mathbb{R}^M$ is drawn from a Gaussian distribution. It holds that $\phi(\mathbf{x}) = [\phi_i(\mathbf{x})]_{i=1}^M$ and $\phi_i(\mathbf{x}) \sim \cos(\mathbf{w}_i^T \mathbf{x} + b_i)$, with $b_i \sim \mathcal{U}[0, 2\pi]$ and $\mathbf{w}_i$ sampled from the kernel function's spectral density [44]. For the Matèrn 5/2 kernel, $\mathbf{w}_i$ are sampled from a d-dimensional t-distribution $\mathbf{w}_i \sim T(0, \Lambda, 5/2)$, with Λ being a diagonal matrix of the GP's kernel lengthscales. For posterior sampling, $\boldsymbol{\theta}$ is sampled from a Gaussian distribution with mean and covariance matrix

$$\begin{aligned} \mu_{\theta|D} &= \left(\Phi^T \Phi + \sigma_n^2 \mathbf{I}\right)^{-1} \Phi^T \mathbf{y} \\ \sigma_{\theta|D}^2 &= \left(\Phi^T \Phi + \sigma_n^2 \mathbf{I}\right)^{-1} \sigma_n^2 \end{aligned} \tag{12}$$

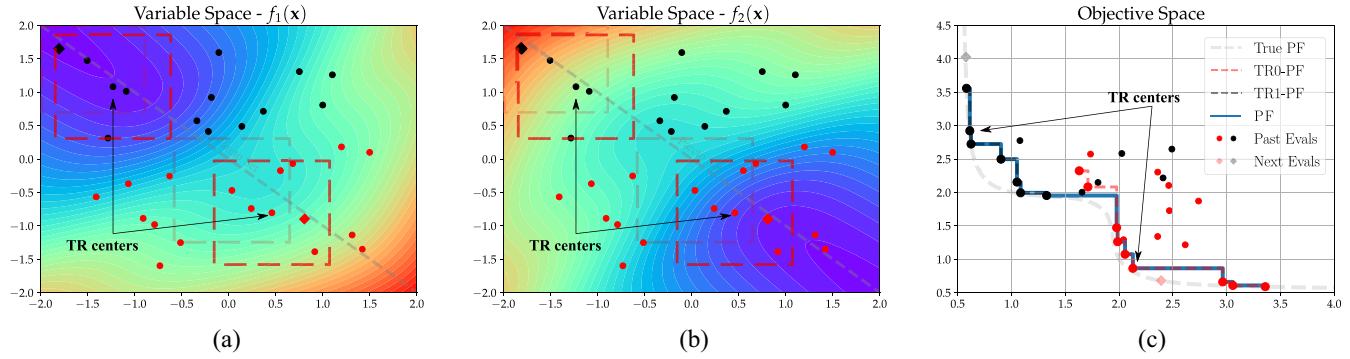

Fig. 1. Demonstration of the local-based approach for minimizing $F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}))]$ in a 2-D space. No constraints apply. (a) and (b) show contour plots of the functions of F. Two trust regions (shown in red boxes) are used, and the sampled points are shown as dots, with the black ones corresponding to the upper trust region. The location of the true PS is shown in light gray, the next query points as diamonds, and the updated trust regions locations are shown as light-red boxes. (c) Shows the mapping to objective space, along with the true PF, the empirical PF, and the local PFs of each trust region.

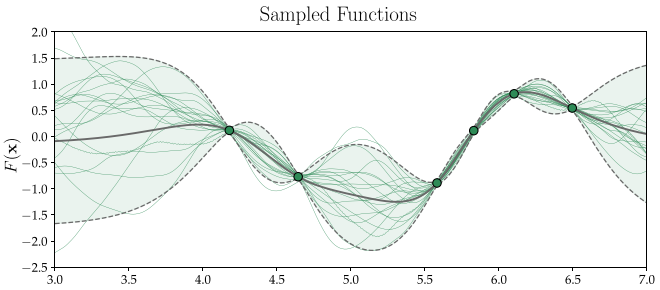

Fig. 2. 20 sampled functions using $M = 300$ RFFs. Past evaluations, GP mean, and 95% confidence bounds are shown.

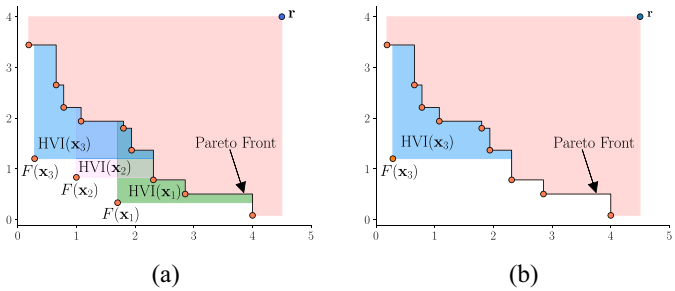

Fig. 3. Illustration of the hypervolume and HVI concepts. Minimization is assumed here. The light-orange area is the dominated HV. Out of three candidates, $\mathbf{x}_3$ provides with the largest HVI. (a) Individual HVI contributions. (b) Total HVI.

where the $(N \times M)$ matrix Φ is given by $\Phi = [\phi(\mathbf{x}_i)]_{i=1}^{N}$ and it holds M features for each of the N sampled points in $\mathcal{D}$. The $(1 \times N)$ vector $\mathbf{y}^T$ contains the past observations from the modeled function. The computational cost for sampling $\tilde{f}(\mathbf{x})$ using the above procedure is $\mathcal{O}(M^3)$ [45].

Fig. 2 shows an example where 20 functions were sampled from a GP using $M = 300$ RFFs. The number of RFFs is a hyperparameter of the overall procedure and it is the same for all employed models. For the rest of this article, M is considered fixed and equal to 300. The readers are referred to [44] for theoretical foundations on RFFs approximations.

Here, functions are sampled to select query points in the constrained multiobjective setting, where query points are restricted to lie in small regions in the variable space.

Unconstrained Problem: Assume for now that there are no constraints and a single trust region is used, with the space occupied by its hypercube denoted by $\mathbb{S}_n^{(1)}$. For each of the trained GP models that approximate the objective functions, one can draw a sample using RFFs. This results in a set of fast-to-evaluate functions $\tilde{F} = [\tilde{f}_1(\cdot), \ldots, \tilde{f}_m(\cdot)]$, which are used in a cheap auxiliary optimization problem

$$\min \ \tilde{F}(\mathbf{x}) = \left[\tilde{f}_1(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x})\right], \quad \mathbf{x} \in \mathbb{S}_n^{(1)}. \tag{13}$$

This problem is solved using NSGA-II [41]. The resulting PS points are query point candidates and are denoted as X_{cand}.

Selecting a query point in X_{cand} requires a metric (measure) of the *utility* of the points in X_{cand}. In our case, where multiple objectives apply, the hypervolume indicator (HV) [46] is an appropriate measure. Following the definition in [46], for a point set $\mathcal{P}$ and a reference vector $\mathbf{r} = [r_1, \ldots, r_m] \in \mathbb{R}^m$, the HV of $\mathcal{P}$ is the m-dimensional Lebesgue measure λ_m of the region that is dominated by $\mathcal{P}$ and bounded by $\mathbf{r}$, i.e.,

$$\text{HV}(\mathcal{P}, \mathbf{r}) = \lambda_m \left(\bigcup_{\mathbf{p} \in \mathcal{P}} [\mathbf{p}, \mathbf{r}] \right). \tag{14}$$

Moreover, HV is used to define the hypervolume improvement (HVI) measure of the PF, used to evaluate new samples. Given a new set of points $\mathcal{Y}$, it is

$$\text{HVI}(\mathcal{P}, \mathcal{Y}, \mathbf{r}) = \text{HV}(\mathcal{P} \cup \mathcal{Y}, \mathbf{r}) - \text{HV}(\mathcal{P}, \mathbf{r}). \tag{15}$$

We select the single query point in X_{cand} as one with the largest HVI when added to the current PF, i.e.,

$$\mathbf{x}^\star = \underset{\mathbf{x} \in X_{\text{cand}}}{\text{argmax}} \, \text{HVI}\big(\mathcal{P}, \tilde{F}(\mathbf{x}), \mathbf{r}\big). \tag{16}$$

Note that the values of the sampled functions are used to evaluate HVI. Also, the reference vector $\mathbf{r}$ is specified either using domain knowledge, or using past evaluations in $\mathcal{D}$, as

$$\mathbf{r} = \left[\max_{\mathbf{x} \in \mathcal{D}}(f_1(\mathbf{x})), \ldots, \max_{\mathbf{x} \in \mathcal{D}}(f_m(\mathbf{x})) \right]. \tag{17}$$

Definition (17) assumes a minimization problem as in (1). If the problem is a maximization one, then the max function should be replaced by min in (17). An illustration of the HV concept is given in Fig. 3.

Constrained Problem: The above procedure extends to constrained problems. In our approach, each constraint function is modeled separately by a GP model. Sampling from all of the employed GP models using RFFs results in an additional set of constraint functions $[\tilde{g}_j(\cdot)]_{j=1}^{l}$. Now, NSGA-II is equipped

with the feasibility rule [35] to account for constraints and the cheap optimization problem is formulated as

$$\begin{aligned} \min \;\; & \tilde{F}(\mathbf{x}) = \left[\tilde{f}_1(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x})\right] \\ \text{s.t.} \;\; & \tilde{g}_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, l, \quad \mathbf{x} \in \mathbb{S}_n^{(1)}. \end{aligned} \tag{18}$$

To select a single query point, we distinguish two cases.

1) If no feasible Pareto optimal solution is found, the set of the candidate points for evaluation, X_{cand}, is formed by the individuals of the final NSGA-II generation, and the query point is selected according to minimize the constraint violation, i.e.,

$$\mathbf{x}^{\star} = \underset{\mathbf{x} \in X_{\text{cand}}}{\operatorname{argmin}} \tilde{\text{CV}}(\mathbf{x}). \tag{19}$$

This drives the optimization toward regions of the search space with low constraint violation. Note that the notation $\tilde{\text{CV}}$ is used to highlight the fact that the constraint violation function is applied on the sampled function values.

2) If feasible Pareto optimal solutions exist, the query point is selected among them, based on their respective HVI contributions.

Multiple query points can be selected by optimizing on multiple GP posterior samples. In this case, N_S batches of objective and constraint functions $[\tilde{F}^{(i)}]_{i=1}^{N_S}$ and $[\tilde{g}_1^{(i)}, \ldots, \tilde{g}_l^{(i)}]_{i=1}^{N_S}$ are sampled and used in N_S separate auxiliary optimization problems as in (18), resulting in N_S batches of candidate points $[X_{\text{cand}}^{(i)}]_{i=1}^{N_S}$. The main challenge is to determine a set of N_S distinct candidate vectors, jointly resulting in maximum HVI. This is a nontrivial task since computing the joint HVI scales exponentially with N_S [46].

Instead, we adopt an iterative greedy approach where no joint HVI calculation is performed. Starting from the current PF HV, the candidate point with maximum HVI contribution from the first batch of sampled functions is selected and added to the PF. The next query point will be selected among the auxiliary optimization results of the next batch of sampled functions using the augmented PF from the previous selection. After N_S iterations, a total of N_S query points are selected for evaluation.

When $N_{TR} > 1$ trust regions are employed, each one holds N_S batches of sampled functions and auxiliary optimization results that reside in their separate hypercube. Similar to the above procedure, N_S query points are selected greedily from N_S augmented candidate pools $[X_{\text{cand}}^{(i)}]_{i=1}^{N_S}$, each one corresponding to one RFF sample. In this case, however, each $X_{\text{cand}}^{(i)}$ is the union of candidate points of all trust regions, based on their ith sample, i.e., $X_{\text{cand}}^{(1)}$ holds the results from the first sample of all trust regions and so on.

Algorithm 1 summarizes the selection scheme for the proposed LoCoMOBO algorithm.

Algorithm 1: Proposed Acquisition Function

Input : batch size N_S; trust region count N_{TR}; GP models $[f_1^{(i)}, \ldots, f_m^{(i)}]_{i=1}^{N_{TR}}$, $[g_1^{(i)}, \ldots, g_l^{(i)}]_{i=1}^{N_{TR}}$; data $\mathcal{D}$, spaces $\{\mathbb{S}_n^{(i)}\}_{i=1}^{N_{TR}}$, reference vector $\mathbf{r}$
Output: query points $\mathbf{x}_i^{\star}$, $i = 1, \ldots, N_S$

```
Compute PF P from D
X_cand^(i) ← ∅ ∀i ∈ {1, ..., N_S}
for i = 1, ..., N_TR do
    for j = 1, ..., N_S do
        Sample f̃_r ~ f_r^(i) using RFF ∀r ∈ {1, ..., m}
        Sample g̃_r ~ g_r^(i) using RFF ∀r ∈ {1, ..., l}
        X_cand ← candidates from problem in (18) on S_n^i
        X_cand^(j) ← X_cand^(j) ∪ X_cand
    end
end
for i = 1, ..., N_S do
    A ← {x ∈ X_cand^(i) | C̃V(x) = 0}
    if A ≠ ∅ then
        x_i^⋆ ← Maximum HVI(P, F̃(x), r) among A
        P ← P ∪ F̃(x_i^⋆)
    else
        x_i^⋆ ← argmin_{x ∈ X_cand^(i)} C̃V(x)
    end
end
```

C. Implementation

Here, we provide implementation instructions for LoCoMOBO and relevant comments. The probabilistic nature of GP models is useful for fast convergence to optimal solutions, but it comes at a high computational cost. In practice, GP inference and training scale cubically with the number of samples used for training [30]. Therefore, as the optimization process progresses and more query points are evaluated, the time spent on GP model training increases. Another important aspect concerns problems with many objective and/or constraint functions; training separate GP models for each one of them can be time consuming and even surpass the time needed for expensive exact function evaluation.

To this end, we leverage recent advances in GP inference by using black-box matrix–matrix multiplication (BBMM) [47]. This uses a highly parallelized routine for matrix–matrix multiplications to perform all computations necessary for GP inference, reducing its asymptotic complexity to $\mathcal{O}(N^2)$ [47]. `GPyTorch` provides a framework for BMMM-based GP training and was used for the development of the GP models. By using tensorial representation for GP kernel matrices, `GPyTorch` allows for the simultaneous training of multiple GP models with considerable speed enhancements [47]. Furthermore, `GPyTorch` enables the use of GPUs for fast inference. In our case, models associated with a single trust region are trained simultaneously.

D. Summary

The complete flow of LoCoMOBO is shown in Algorithm 2. The maximum number of iterations is computed by the maximum number of evaluations and the batch size N_S. It is worth mentioning that the GP models of a trust region are trained only when new query points coming from the same trust region are selected and evaluated.

To demonstrate the benefits of LoCoMOBO, we consider a set of benchmark functions for constrained MOPs with varying dimensions. OSY is a 6-D problem having 2 objectives and 6 constraints, MW2 has 15 variables, 2 objectives, and 1 constraint, and C2DTLZ2 has 12 variables, 3 objectives, and 1 constraint. For performance evaluation, we use the HV metric of the resulting PF, with vector $\mathbf{r}$ equal to [0, 100], [1.5, 1.5], and [1.1, 1.1, 1.1], respectively.

To highlight the advantages of the local-based approach, we introduce two baseline methods for comparison.

Algorithm 2: LoCoMOBO Algorithm

Input : trust region count N_{TR}; batch size N_S, initial samples N_0, maximum iterations T_{max}, initial trust region lengths
Output: PF, PS

Evaluate objectives and constraints at N_0 random points
Initialize $\mathcal{D}_i$ $\forall i \in \{1, \ldots, N_{TR}\}$ with all initial samples
k-Means clustering of initial samples and select $\mathbf{x}_{c,i}$ $\forall i \in \{1, \ldots, N_{TR}\}$ as the i-th centroid
for $i = 1, \ldots, T_{max}$ **do**
    Select N_S query points using the acquisition function, (no black-box function evaluation at this step)
    Evaluate objectives and constraints at selected points
    for $j = 1 \ldots, N_{TR}$ **do**
        if j-th trust region evaluated query points **then**
            Update $\mathcal{D}_j$ and trust region center $\mathbf{x}_{c,j}$
            Train GP models of j-th trust region
        end
        Update $\mathcal{L}_j$
    end
end

TABLE I
HV RESULTS (MEAN $\pm$ STD) ON BENCHMARK FUNCTIONS

	OSY	MW2	C2DTLZ2
LoCoMOBO N_{TR}=1	$\mathbf{21339 \pm 345}$	1.34 ± 0.11	0.49 ± 0.02
LoCoMOBO N_{TR}=2	20956 ± 616	$\mathbf{1.46 \pm 0.12}$	$\mathbf{0.54 \pm 0.06}$
WEIBO	10343 ± 1207	NA	0.283 ± 0.12
MESMOC	11394 ± 977	NA	0.184 ± 0.02
Baseline-1	21056 ± 522	0.92 ± 0.09	0.46 ± 0.03
Baseline-2	20097 ± 609	0.81 ± 0.12	0.43 ± 0.05
NSGA-II	11508 ± 3105	1.12 ± 0.22	0.39 ± 0.11
NSGA-III	12213 ± 1793	1.22 ± 0.15	0.45 ± 0.03

1) Baseline-1 uses the proposed acquisition function with a single trust region that spans the entire variable space. This is a special case of the proposed algorithm and can be considered as a constraint handling TSEMO [38] variant.
2) Baseline-2 uses a single trust region that also spans the entire variable space, and selects query points based on the PF of LCB acquisition functions on each objective, such as in [33]. To account for constraints, the pointwise values of each LCB function are weighted by their probability of feasibility [28]. The most diverse candidate solutions are selected as query points from the resulting PF.

For the proposed approach, we include a single-trust-region (LoCoMOBO $N_{TR} = 1$) and a two-trust-region (LoCoMOBO $N_{TR} = 2$) approach, both of which use local trust regions, unlike Baseline-1. The batch size for the above methods is set to $N_S = 5$. WEIBO [28], MESMOC [32], NSGA-II, and NSGA-III [48] are also included for comparisons. For all BO-related methods, the initial samples were set to $2(d+1)$ where d is the dimension of each problem, and the maximum evaluations were set to 200 for OSY, 500 for C2DTLZ2, and 900 for the demanding MW2. Maximum evaluations for NSGA-II and NSGA-III are set to three times the aforementioned limits, and the number of reference directions is 24.

Table I shows the results of the experiments. LoCoMOBO outperforms the other methods in the given evaluation budgets. For the case of OSY, it is evident that a single trust region is enough to produce acceptable results. This can be verified by the performance of global Baseline-1. For the other two functions, the two-trust-region LoCoMOBO performs better than the single trust-region one, while the rest of the methods

TABLE II
RUNTIME COMPARISON (S)

		OSY	MW2	C2DTLZ2
GPU	LoCoMOBO N_{TR}=1	57	531	184
	LoCoMOBO N_{TR}=2	73	706	251
	WEIBO	944	4671	1832
CPU	LoCoMOBO N_{TR}=1	65	1887	345
	LoCoMOBO N_{TR}=2	77	2217	393
	WEIBO	1124	15933	3519

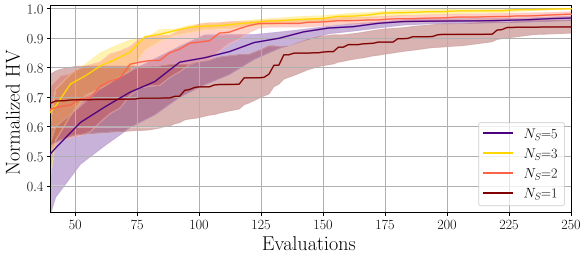

Fig. 4. Evolution of HV attained by the proposed approach using different batch sizes. The values in the y axis are normalized to the largest HV value attained.

either fail to find feasible points or result in much worse PFs. This is demonstrated in the case of MW2, where the feasible region occupies little portion of the design space, showing that the proposed method can handle effectively constrained problems. The multiple trust-region approach proves to be effective in high-dimensional problems, where Pareto optimal solutions may lie away from each other or in disconnected regions.

To provide a quantitative measure of the speed gains of GPU acceleration, Table II shows the average runtimes of the above experiments for the batched proposed LoCoMOBO and the sequential WEIBO. It is shown that a slight increase in runtime is induced by including additional trust regions, which is expected as more GP models need to be trained. As expected, the batched approach is faster than the sequential WEIBO, since GP model training is done fewer times for the same sample budget. In the 15-D MW2 problem, where the sample budget is larger compared to the other experiments, the inclusion of GPU acceleration provides a speedup of $\times 3.5$ in the optimization procedure.

By optimizing the OSY function using different batch size N_S, we demonstrate the effect of the batch size parameter on the LoCoMOBO's performance. Fig. 4 shows the results of this experiment, highlighting that there is no profound difference in the performance of LoCoMOBO with respect different batch sizes. In this case, we used a single trust region. We have found out empirically that LoCoMOBO provides better results when using a batch size in the range $[5, 10]$ for MOPs with more than 20 variables. It is worth noting that every experiment of this section was repeated five times to account for random fluctuations.

IV. CIRCUIT DESIGN APPLICATIONS

In this section, LoCoMOBO is applied to three real-world circuits, two CMOS amplifiers, and an LNA to demonstrate its effectiveness. To highlight that the proposed approach can handle multiple objectives as well as process, voltage, and temperature (PVT) variations, we consider the following formulations for each circuit. Both CMOS amplifiers are sized accounting only for their nominal performance, using a

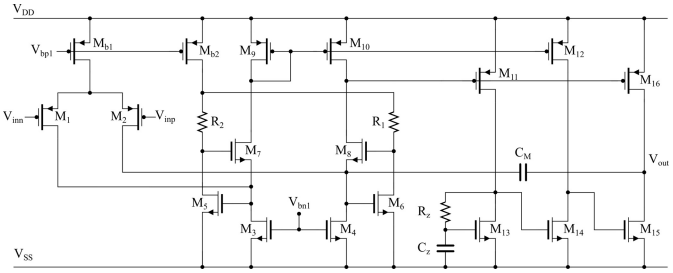

Fig. 5. Three stage amplifier proposed in [49].

two-objective and a three-objective formulation. In the case of the LNA, we consider two objectives and include additional constraints to account for five different process corners, three different operating temperatures, and three supply voltages. All of the discussed circuits are designed in Cadence Virtuoso and Spectre is used for simulation, using a TSMC 90-nm PDK.

The algorithms discussed are implemented in Python, using `GPyTorch` and the `DEAP` library. The available online code for MESMOC [32] was used as well. All of the experiments were executed on a Linux workstation using an 8 core CPU and a P5000 GPU.

A. Three Stage Amplifier

A three stage amplifier [49] shown in Fig. 5 is sized in this section. It includes a wideband current buffer and an active left-half-plane zero [49] to increase its driveability to large capacitive loads, without sacrificing its bandwidth.

The amplifier consists of 18 transistors, the halves of the current mirrors generating voltages V_{bn1} and V_{bp1} (not shown in Fig. 5), three resistors, and two capacitors. Based on the circuit's topology and taking into account symmetry constraints, a total of 23 independent variables are used for sizing. These include 13 transistor widths, four transistor lengths, two biasing currents, and R_1, R_z, C_z, and C_M. Assuming no prior knowledge of the optimal parameter values, the ranges of the transistors' lengths and widths are restricted to be less than 1.5 and 100 μm, respectively, while their lower bounds are equal to the minimum acceptable by the PDK. The allowable ranges for capacitors, resistors and biasing currents are $[0.5, 6]$ pF, $[0.1, 300]$ kΩ, and $[0.4, 6]$ μA, respectively. It should be noted that any restriction on the silicon area can be enforced by defining appropriate variable ranges. In addition, the in-house interface tool used allows the definition of separate variable ranges for each device of the circuit studied. To obtain the circuit's performance metrics, two testbenches are used; one for transient simulation and slew rate measurements, and the other for ac and dc analysis.

1) Two-Objective Formulation: For the two-objective case, we seek to determine the optimal tradeoff between power consumption and dc voltage gain. The remaining specifications, based on the original implementation in [49] with the capacitive load of $C_L = 15$ nF, constitute the optimization constraints and are given in detail in Table III.

For LoCoMOBO, we consider two cases with a single and two trust regions operating locally in the search space. All BO-related methods discussed in the previous section are applied for comparison. The maximum number of simulations is 1300 and the initial sample count is 150. Two batch sizes $N_S = 5$ and $N_S = 10$ are used for LoCoMOBO and the two baseline methods. In this experiment, NSGA-II has population 50 and 80 generations. All experiments are repeated ten times to account for random fluctuations and the results in terms of PF HV are given in Table IV.

TABLE III
SPECIFICATIONS FOR THREE STAGE AMPLIFIER—TWO OBJECTIVES

Performance Metrics	Specifications [49]
DCGain	maximize
P_{dc}	minimize
SR_{avg}	$\geq 0.22V/\mu s$
UGF	≥ 0.95MHz
PM	$\geq 52.3^o$
GM	≥ 18dB
C_L	$= 15$nF
V_{DD}	$= 2$V

TABLE IV
THREE STAGE AMPLIFIER HV RESULTS—TWO OBJECTIVES

		HV			Runtime (min)
		Mean	Best	Worst	Mean
N_S=5	LoCoMOBO N_{TR}=1	2147	2243	1928	60
	LoCoMOBO N_{TR}=2	2165	2206	1958	70
	Baseline-1	1481	1589	1385	61
	Baseline-2	1597	1755	1298	59
N_S=10	LoCoMOBO N_{TR}=1	2014	2158	**1984**	28
	LoCoMOBO N_{TR}=2	**2187**	**2284**	1975	32
	Baseline-1	1512	1754	1414	29
	Baseline-2	1661	1994	1387	28
	WEIBO	1262	1420	1136	654
	MESMOC	926	1003	859	1171
	NSGA-II	1126	1285	921	241
	NSGA-III	1325	1456	1207	259

We used the worst performance values from the PFs of all algorithms and runs to determine the reference point, in a similar manner as in (17), and utilized the total biasing current instead of power consumption as an indicator. This resulted in $\mathbf{r} = [81.9, 321/V_{DD}] = [81.9, 160.5]$. Except for MESMOC, WEIBO, and NSGA-II, all algorithms take advantage of batched simulation execution. The GP models of LoCoMOBO and the baseline methods were trained using both GPU acceleration and batch GP training, and WEIBO models were trained using only GPU acceleration.

In the case of $N_S = 5$ and $N_{TR} = 1$, LoCoMOBO provides a runtime gain of $\times 10.9$ and $\times 19.6$ compared to WEIBO and MESMOC, respectively. For $N_S = 10$, the aforementioned speedup is by $\times 23$ and $\times 42$, respectively. The two-trust-region optimization takes $\times 1.2$ more time to complete compared to the single-trust-region case, due to the training of additional GP models.

As shown in Table IV, in terms of dominated HV, it is evident that LoCoMOBO finds better solutions within the simulation budget restrictions. The two-trust-region approach provides on average the best PFs in the case of $N_S = 10$. LoCoMOBO consistently surpasses the examined algorithms considering different batch sizes and trust-region count.

During the experiments, we noticed that the main difficulty when sizing this topology is that the optimum solutions lie close to the infeasible region. This is the reason why the entropy-based MESMOC results in lesser HV than the rest of the methods, since it found only a small amount of feasible solutions. An interesting note is that the batched methods provide better results compared to the sequential ones. The PFs resulting from a single run per algorithm are given for demonstration in Fig. 6.

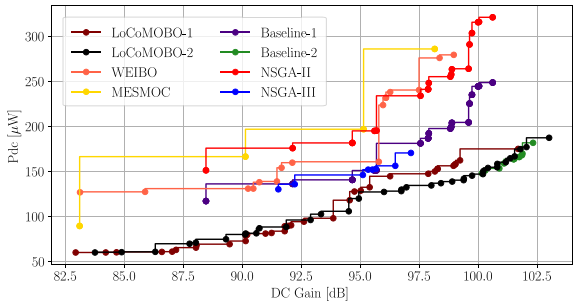

Fig. 6. PFs for the three stage amplifier. Bottom-right values result in better dc gain versus Pdc tradeoff.

TABLE V
THREE STAGE AMPLIFIER COVER MATRIX—TWO OBJECTIVES

	LoCoMOBO N_{TR}=1	LoCoMOBO N_{TR}=2	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO N_{TR}=1	-	0.25	1	1	1	0.04	1	1
LoCoMOBO N_{TR}=2	0.44	-	1	1	1	0.08	1	1
WEIBO	0	0	-	0.75	0	0	0.32	0
MESMOC	0	0	0	-	0	0	0	0
Baseline-1	0	0	0.81	0.75	-	0	1	0.37
Baseline-2	0.23	0.17	0.63	0.75	0.78	-	1	0.63
NSGA-II	0	0	0.25	0.5	0	0	-	0
NSGA-III	0	0	0.67	0.5	0.05	0	0.32	-

To demonstrate in a qualitative manner the results of the experiments, we provide a cover matrix [28] in Table V. This is a matrix where its (i, j) entry represents the portion of PF solutions from the jth algorithm that are dominated by the PF solutions of the ith one, providing an additional metric for comparison between two different PFs.

We used the PFs from the $N_S = 10$ case to compute the cover matrix. It is seen that the PFs from LoCoMOBO dominate most of the Pareto solutions resulting from the rest of the examined approaches. The results from the two-trust region optimization dominate 44% of the ones from the single trust region optimization. Among the rest of the methods, Baseline-2 provides better PFs qualitatively. It provides a dense set of PF solutions, which explains the fact that its PF is not dominated by any other approach, but it does not yield diverse solutions such as LoCoMOBO. For the single-trust-region LoCoMOBO PF of Fig. 6, the average, maximum, and minimum typical FOM$_S$ [49] of the PF solutions are 105, 113, and 99 (MHz·pF/μW). The same metrics for NSGA-II are 96, 104, and 85.

2) Three-Objective Formulation: We now proceed with the formulation of the three-objective optimization problem. The design space remains the same, and the optimization goals and constraints are given in Table VI. In this case, the simulation budget for BO methods is increased to 1600 and the initial sample count to 150. For NSGA-II, 50 individuals and 100 generations are used. Again, all experiments were executed ten times to account for random effects.

The results in terms of final PF HV are given in Table VII, with $\mathbf{r} = [80.3, 473, 0.05]$. LoCoMOBO outperforms the other algorithms in this case as well. By using two trust regions, we are able to get slightly better results. In this case, LoCoMOBO performs better when using a smaller batch size. Among the rest of the methods, Baseline-2 results in the best

TABLE VI
SPECIFICATIONS FOR THREE STAGE AMPLIFIER—THREE OBJECTIVES

Performance Metrics	Specifications [49]
DCGain	maximize
P_{dc}	minimize
SR_{avg}	maximize
UGF	≥ 0.95MHz
PM	$\geq 52.3^o$
GM	$\geq$ 18dB
C_L	= 15nF
V_{DD}	= 2V

TABLE VII
THREE STAGE AMPLIFIER HV RESULTS—THREE OBJECTIVES

		HV			Runtime (min)
		Mean	Best	Worst	Mean
N_S=5	LoCoMOBO N_{TR}=1	7983	8104	7475	73
	LoCoMOBO N_{TR}=2	**8034**	**8164**	**7606**	80
	Baseline-1	5646	5881	5354	73
	Baseline-2	6046	6573	5319	71
N_S=10	LoCoMOBO N_{TR}=1	7752	8127	7254	42
	LoCoMOBO N_{TR}=2	7821	8095	7485	45
	Baseline-1	5366	5576	5256	41
	Baseline-2	6458	6808	5985	41
	WEIBO	4270	4786	3864	825
	MESMOC	4897	5576	4668	1528
	NSGA-II	2809	3315	2633	308
	NSGA-III	4348	4779	3954	321

TABLE VIII
THREE STAGE AMPLIFIER COVER MATRIX—THREE OBJECTIVES

	LoCoMOBO N_{TR}=1	LoCoMOBO N_{TR}=2	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO N_{TR}=1	-	0.21	0.87	0.85	0.86	0.79	1	0.79
LoCoMOBO N_{TR}=2	0.28	-	1	0.9	0.96	0.97	1	1
WEIBO	0.01	0	-	0.1	0.03	0.08	0	0.1
MESMOC	0.02	0.02	0.38	-	0.12	0.28	0.63	0.21
Baseline-1	0.02	0.01	0.66	0.54	-	0.1	0.84	0.52
Baseline-2	0.11	0.06	1	0.4	0.7	-	0.95	1
NSGA-II	0	0	0.02	0.02	0	0	-	0
NSGA-III	0	0	0.11	0.08	0.03	0	0.83	-

solutions. The fact that LoCoMOBO consistently outperforms the global Baseline-1 highlights the advantage of the local-based approach. In terms of runtime comparison, the proposed approach is $\times 20$ and $\times 36$ faster compared to the sequential WEIBO and MESMOC, respectively, for $N_S = 10$.

The PFs resulting from the three objectives formulation ($N_S = 10$) are compared qualitatively in the cover matrix of Table VIII.

Here, it is seen that LoCoMOBO dominates most of the solutions of the other approaches, as expected from the HV results. In this problem, most of the algorithms found feasible solutions from the initial samples; therefore, the ratio of feasible to infeasible space is quite large. This explains the fact that some approaches performed better in the three-objective case compared to the two-objective one. For demonstration purposes, an example PF resulting from LoCoMOBO ($N_{TR} = 2$ and $N_S = 10$) is shown in Fig. 7.

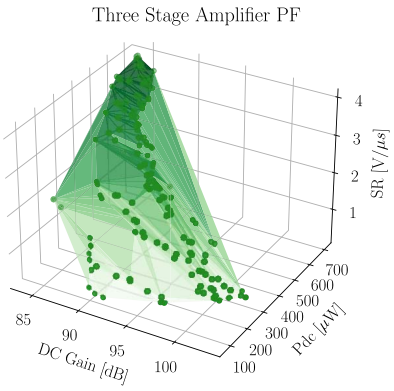

Fig. 7. PF for the three stage amplifier.

TABLE IX
SPECIFICATIONS FOR FOUR STAGE AMPLIFIER—TWO OBJECTIVES

Performance Metrics	Specifications [50]
DCGain	maximize
P_{dc}	minimize
SR_{avg}	$\geq 0.14V/\mu s$
UGF	≥ 1.18MHz
PM	$\geq 48.1^o$
GM	≥ 8.26dB
V_{DD}	$= 1.2$V
C_L	$= 12$nF

TABLE X
FOUR STAGE AMPLIFIER HV RESULTS—TWO OBJECTIVES

		HV			Runtime (min)
		Mean	Best	Worst	Mean
N_S=5	LoCoMOBO N_{TR}=1	15204	15866	14788	67
	LoCoMOBO N_{TR}=2	17299	17425	**16897**	81
	Baseline-1	14019	14565	13843	65
	Baseline-2	13462	14187	12892	64
N_S=10	LoCoMOBO N_{TR}=1	15515	15725	15212	31
	LoCoMOBO N_{TR}=2	**17416**	**17854**	16680	44
	Baseline-1	14418	14810	13982	30
	Baseline-2	15014	15794	14424	32
	WEIBO	11943	12384	11205	754
	MESMOC	11702	11918	11399	1333
	NSGA-II	10942	12075	9547	305
	NSGA-III	11639	1285	921	312

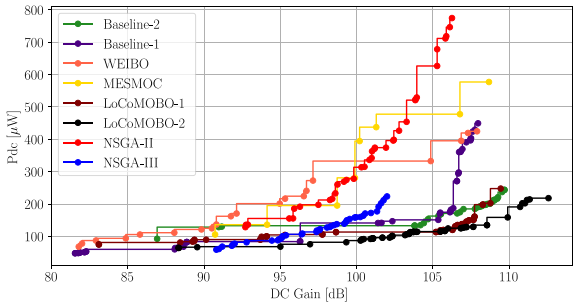

Fig. 9. PFs for the four stage amplifier. Bottom-right values result in better dc gain versus Pdc tradeoff.

B. Four Stage Amplifier

A four-stage amplifier [50] shown in Fig. 8 is examined in this experiment. It can drive large capacitive loads by employing an active zero subcircuit, a slew-rate enhancer subcircuit, and four gain stages. Similar to the previous case, we consider two optimization formulations, one with two and one with three objectives, respectively.

This circuit consists of the core amplifier shown in Fig. 8 and a biasing circuit that is responsible for voltages V_{bn1}, V_{bn2}, V_{bp1}, and V_{bp2} [50] (not shown in Fig. 8). In total, 35 transistors, 2 capacitors, a single resistor, and a current source are employed. We use two testbenches, one for the slew rate and one for ac and dc analysis. The testbenches are parameterized by 43 parameters, including 20 transistor widths, 19 transistor lengths, a bias current, and C_Z, C_M, and R_Z. The variable ranges are determined as follows. Transistor lengths and widths are again restricted to be less than 1.5 and 100 μm, respectively, with the lower bounds set to the minimum acceptable values of the PDK. The biasing current range is $[0.5, 10]$ μA and the ranges of the resistor and the capacitors are the same as in the previous example.

1) Two-Objective Formulation: The objectives and constraints for the two-objective formulation are based on the original implementation in the case of $C_L = 12$ nF and they are given in Table IX. For the BO-related methods, the total simulation budget is 1300, with 150 initial sampling simulations. For NSGA-II, the population count is 60 and the number of generations is 80.

In terms of HV, the optimization results are given in Table X.
These were computed after ten repetitions of each experiment, with the reference point $\mathbf{r} = [81, 645]$ determined using the worst PF values among all algorithms and executions. The gains from the two-trust-region LoCoMOBO are more evident in this higher dimensional case. Among all methods, the two-trust-region LoCoMOBO with $N_S = 10$ delivers the best PF with a slight overhead in runtime ($\times 1.4$ slower) compared to the single-trust-region LoCoMOBO.

In this problem, the single-trust-region LoCoMOBO is $\times 24$ and $\times 43$ faster than the sequential WEIBO and MESMOC when using $N_S = 10$. Therefore, LoCoMOBO provides better PFs within shorter time limits compared to the other algorithms. The PFs resulting from the optimization procedures ($N_S = 10$) are shown in Fig. 9. LoCoMOBO $N_{TR} = 1$ has average, maximum, and minimum FOM$_S$ 105, 166, and 64 (MHz·pF/μW) in its PF of Fig. 9. For comparison, NSGA-II provides with 85, 125, and 49.

We compare the results ($N_S = 10$) in terms of Pareto dominance using the cover matrix shown in Table XI. In this case, where the variable space is larger compared to the previous example, the two-trust region LoCoMOBO dominates 88% of the PF solutions from the single trust-region one. Both proposed approaches dominate large portions of the PFs from the rest of the methods. Baseline-2 again proves favorable in comparison to Baseline-1, WEIBO, MESMOC, NSGA-II, and NSGA-III.

2) Three-Objective Formulation: The variable space remains the same in the three-objective formulation and the objectives and constraints are given in Table XII. Here, the total simulation budget for the BO-related methods is increased to 1600, with 150 initial sampling simulations. The NSGA-II population count and maximum generations are 60 and 100, respectively. The results in terms of final PF HV are given in Table XIII, with $\mathbf{r} = [75, 497, 0.027]$.

The proposed approach provides higher HV values compared to those of all the other approaches. The two-trust-region

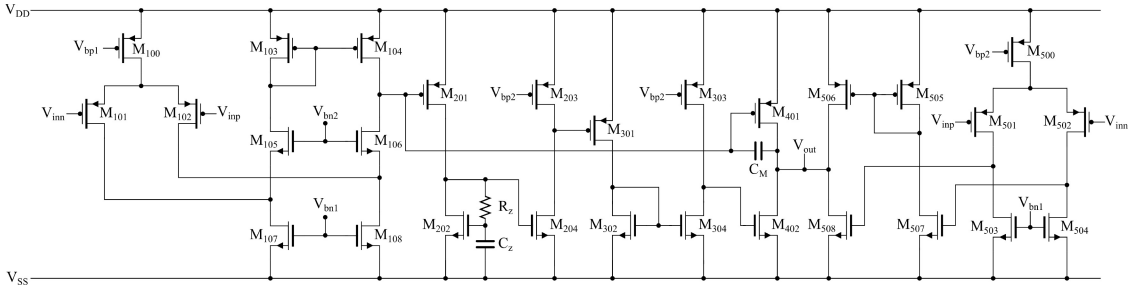

Fig. 8. Four stage amplifier proposed in [50].

TABLE XI
FOUR STAGE AMPLIFIER COVER MATRIX—TWO OBJECTIVES

	LoCoMOBO N_{TR}=1	LoCoMOBO N_{TR}=2	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO N_{TR}=1	-	0.12	0.95	1	0.56	0.43	1	0.68
LoCoMOBO N_{TR}=2	0.88	-	1	1	0.72	1	1	0.95
WEIBO	0	0	-	0.38	0.02	0	0.53	0.05
MESMOC	0	0	0.55	-	0	0	0.38	0
Baseline-1	0.28	0	0.95	0.87	-	0.1	1	0.53
Baseline-2	0.06	0	0.7	0.88	0.54	-	0.97	0.43
NSGA-II	0	0	0.4	0.25	0	0.05	-	0
NSGA-III	0.16	0.05	0.65	0.75	0.17	0.08	0.63	-

TABLE XII
SPECIFICATIONS FOR FOUR STAGE AMPLIFIER—THREE OBJECTIVES

Performance Metrics	Specifications [50]
DCGain	maximize
P_{dc}	minimize
SR_{avg}	maximize
UGF	≥ 1.18MHz
PM	$\geq 48.1^o$
GM	≥ 8.26dB
V_{DD}	$= 1.2$V
C_L	$= 12$nF

LoCoMOBO again proves favorable compared to the single-trust-region one, but with the outcome of higher variance. This is highlighted by the fact that the single-trust-region LoCoMOBO provides the least worst solution among all algorithms. We noticed that all algorithms managed to find feasible solutions with relative ease, which means that this problems is not highly constrained. This may explain the fact that WEIBO and MESMOC provide better solutions compared to their two-objective performances, but they do not manage to come close to the HV values attained by LoCoMOBO. Selecting and evaluating $N_S = 5$ query points prove favorable in this case. We argue that this is due to the multimodal loss landscape of this problem; GP models need to be trained more often before providing sampled functions to the acquisition function.

In terms of runtime, the single-trust-region LoCoMOBO with $N_S = 5$ provides an overall speedup of $\times 12$ and $\times 21$ compared to WEIBO and MESMOC, respectively. These quantities become $\times 20$ and $\times 35$ when the batch size is $N_S = 10$. A qualitative comparison of the resulting PFs from the tested methods ($N_S = 10$) is given in the cover matrix of Table XIV.

The PFs of LoCoMOBO dominate a large portion of the PFs of the other algorithms. In addition, the two-trust-region case seems to provide with wider PFs in comparison to the single-trust-region one. While it provides a relatively large PF HV difference, it does not dominate large portion of the single-trust-region PF. The HV difference, therefore, can be attributed to more diverse PF. An example PF for this circuit ($N_{TR} = 2$ and $N_S = 10$) is given in Fig. 10.

TABLE XIII
FOUR STAGE AMPLIFIER HV RESULTS—THREE OBJECTIVES

		HV			Runtime (min)
		Mean	Best	Worst	Mean
N_S=5	LoCoMOBO N_{TR}=1	4721	4895	**4588**	79
	LoCoMOBO N_{TR}=2	**4745**	**4954**	4512	93
	Baseline-1	3392	3604	3184	80
	Baseline-2	3561	3813	3197	77
N_S=10	LoCoMOBO N_{TR}=1	4321	4502	4150	47
	LoCoMOBO N_{TR}=2	4707	4925	4571	63
	Baseline-1	3276	3539	3095	49
	Baseline-2	3762	3953	3413	47
	WEIBO	3728	4026	3376	951
	MESMOC	3621	3892	3450	1685
	NSGA-II	2700	3106	2503	383
	NSGA-III	3185	2912	3397	396

TABLE XIV
FOUR STAGE AMPLIFIER COVER MATRIX—THREE OBJECTIVES

	LoCoMOBO N_{TR}=1	LoCoMOBO N_{TR}=2	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO N_{TR}=1	-	0.33	0.53	0.69	0.61	0.42	0.75	0.58
LoCoMOBO N_{TR}=2	0.57	-	0.64	0.77	0.81	0.68	0.91	0.76
WEIBO	0.21	0.02	-	0.15	0.26	0.24	0.36	0.06
MESMOC	0.07	0.03	0.2	-	0.25	0.22	0.32	0.07
Baseline-1	0.1	0.07	0.25	0.6	-	0.1	0.53	0.28
Baseline-2	0.17	0.02	0.5	0.18	0.31	-	0.67	0.48
NSGA-II	0.01	0	0.07	0.04	0.06	0.02	-	0
NSGA-III	0.08	0.05	0.17	0.24	0.13	0.11	0.25	-

C. Low Noise Amplifier

An LNA is sized in this section. The circuit is shown in Fig. 11 and it is an inductively degenerated common-source cascode topology.

The LNA consists of three nMOS transistors (one current source), three inductors, three capacitors, and two resistors. All of the employed inductors are spiral ones and capacitors C_g, C_d, and C_f are metal-oxide-metal (MOM) ones. Devices R_{b1} and R_{b2} are p-diffusion resistors, while R_d is fixed to 100Ω. In total, there are 21 independent variables and their ranges are as follows. Inductor widths are chosen between the available values by the PDK (3, 6, 9, and 15 μm), while the number

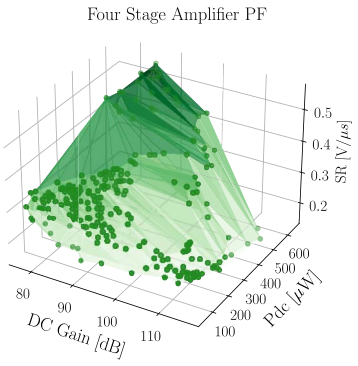

Fig. 10. PF for the four stage amplifier.

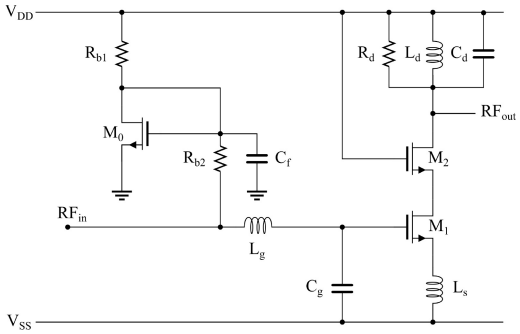

Fig. 11. LNA examined in this section.

of turns ranges between 0.5 and 5 with a step of 0.25 and the inner radius between 10 and 100 μm. Transistor length and width ranges are $[100, 240]$ nm and $[5, 250]$ μm. The total length of the resistors ranges between 1 and 30 μm, while their width is fixed to 2 μm. For the capacitors, the fingers width is fixed to 140 nm and the number of horizontal and vertical fingers ranges between 1 and 100.

This problem involves both continuous and integer-valued variables. To handle this variable space, we adopt the methodology proposed in [51] and extend LoCoMOBO to handle mixed-integer variables. To this end, prior to the evaluation of any sampled function via RFF when solving (18), the input vector $\mathbf{x}$ is replaced by a transformed vector $T(\mathbf{x})$, where $T(\cdot)$ rounds the input entries that correspond to integer variables to the closest integer.

In this experiment, we consider optimizing for power dissipation and gain (S_{21}) with 2.4-GHz operating frequency. To account for PVT variations, we consider 45 separate testbenches, combining five different corner model files (typical, SS, SF, FS, and FF), three operating temperatures (-50, 27, and 125 C), and three supply voltages (typical, typical $\times 1.1$, and typical $\times 0.9$). The specifications for the LNA, which include optimization objectives and constraints, are given in detail in Table XV. In summary, the constraints are chosen to ensure that a 20-dB gain is achieved in typical conditions, while linearity, input matching, and noise performances at all corners meet the chosen thresholds.

In this example, we compare the proposed LoCoMOBO approach using a single and two trust regions against Baseline-1, NSGA-II, and NSGA-II. The batch size for BO approaches is $N_S = 5$. WEIBO and MESMOC were not proposed to handle problems with mixed integer and discrete variables and Baseline-2 cannot be extended to do so, thus they are not included in this test case. For LoCoMOBO and Baseline-1, the initial samples are 150 and the maximum number of evaluations is 1300. For

TABLE XV
SPECIFICATIONS FOR LNA

Performance Metrics	Testbenches	Specifications
S_{21}@2.4GHz	nominal	maximize
P_{dc}	nominal	minimize
S_{21}@2.4GHz	nominal	$\geq$ 20dB
S_{11}@2.4GHz	all	≤ -8dB
S_{22}@2.4GHz	all	≤ -8dB
IIP3	all	≥ -5dBm
NF @2.4GHz	all	$\leq$ 2.5dB

TABLE XVI
LNA HV RESULTS

		HV			Runtime (min)
		Mean	Best	Worst	Mean
N_S=5	LoCoMOBO N_{TR}=1	44.2	45.3	40.8	144
	LoCoMOBO N_{TR}=2	**45.1**	**46.2**	**43.6**	152
	Baseline-1	39.1	42.2	35.8	139
	NSGA-II	22.9	27.2	18.1	1586
	NSGA-III	27.1	35.4	21.8	1590

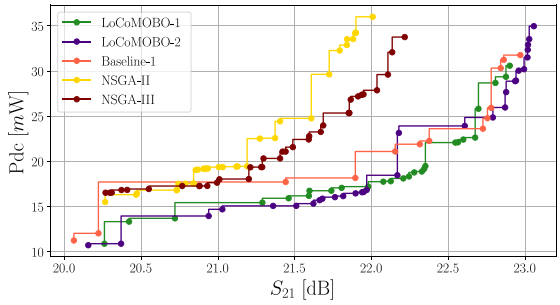

Fig. 12. PFs for the LNA. Bottom-right values result in better S_{21} versus Pdc tradeoff.

NSGA-II and NSGA-III, 50 individuals and 80 generations were used, with 24 reference directions. The experiments are repeated five times to account for random flunctuations and the results in terms of HV are given in Table XVI, where $\mathbf{r} = [20.06, 30]$, computed from the minimum gain and maximum dc current encountered from all of the algorithms.

It is seen that LoCoMOBO outperforms the population-based NSGA-II and NSGA-III in this example as well. The global Baseline-1 performs better than the population-based algorithms but it does not reach the performance of the proposed approach using a single or two trust regions. A cover matrix for this example is shown in Table XVII. Here, the results from the HV-based comparison are validated, demonstrating the effectiveness of the proposed approach. For demonstration purposes, the resulting PFs from a single run of each algorithm are shown in Fig. 12. For the case of LoCoMOBO with a single trust region, the average, maximum, and minimum LNA FOM [52] from the PF solutions are 17485, 21060, and 13069 W^{-1}. For comparison, the same metrics for NSGA-II are 13884, 16233, and 9881.

To highlight the effect of including PVT variations in the optimization procedure, we proceed with a comparison between variation-aware and nominal optimization results. The above experiment is repeated once more, but this time accounting only for nominal conditions; therefore, we utilize a single testbench. The variable ranges as well as the algorithmic hyperparameters remain the same and the optimization constraints are changed accordingly. In Fig. 13, the PFs from the nominal optimization and the ones resulting from the corner

TABLE XVII
LNA COVER MATRIX

	LoCoMOBO N_{TR}=1	LoCoMOBO N_{TR}=2	Baseline-1	NSGA-II	NSGA-III
LoCoMOBO N_{TR}=1	-	0.1	0.69	1	1
LoCoMOBO N_{TR}=2	0.38	-	0.69	1	1
Baseline-1	0.04	0.06	-	0.71	0.69
NSGA-II	0	0	0	-	0.16
NSGA-III	0	0	0	0.84	-

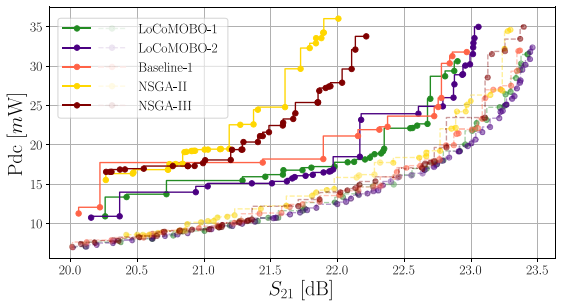

Fig. 13. Comparison between the PFs derived accounting for PVT variations and only for nominal conditions. Nominal sizing results are depicted using light dotted lines and PVT-aware results using solid lines.

based one are superimposed. Two major conclusions can be drawn from this figure: 1) LoCoMOBO outperforms the other algorithms considering both nominal conditions and PVT variations and 2) the PFs from the variation-aware sizing provide worse S_{21} versus P_{DC} tradeoff, which is expected due to the additional constraints that ought to be satisfied. This, in turn, implies that the nominal PFs do not satisfy the constraints of Table XV in all corners.

V. CONCLUSION

A new local multiobjective BO method that addresses constrained problems, LoCoMOBO, was proposed in this article. In contrast to classical BO approaches that search the entire variable space, LoCoMOBO provides Pareto optimal solutions by using a trust-region approach, which maintains separate GP models in the search space. The size and location of the trust regions are adaptively updated throughout the optimization. Using TS and RFFs, a set of auxiliary cheap MOPs is defined and solved to suggest multiple query points at each iteration, thereby allowing for parallel query point evaluation. The efficiency of LoCoMOBO was demonstrated by comparing it against state-of-the-art approaches and two baseline methods on a set of benchmark functions and three real-world circuits. Both nominal conditions and cases accounting for PVT variations were examined to highlight the flexibility of the proposed approach. The experimental results suggest that LoCoMOBO provides better results both in terms of PF quality and total optimization runtime.

REFERENCES

[1] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," *Proc. IEEE*, vol. 95, no. 3, pp. 640–669, Mar. 2007.

[2] K. M. H. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "A 90 nm CMOS, 6 μW power-proportional acoustic sensing frontend for voice activity detection," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 291–302, Jan. 2016.

[3] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums, "Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 6, pp. 703–717, Jun. 2000.

[4] N. Guo *et al.*, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1514–1524, Jul. 2016.

[5] L. B. Leene and T. G. Constandinou, "Time domain processing techniques using ring oscillator-based filter structures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 12, pp. 3003–3012, Dec. 2017.

[6] D. G. Chen, F. Tang, M.-K. Law, and A. Bermak, "A 12 pj/pixel analog-to-information converter based 816 × 640 pixel CMOS image sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 5, pp. 1210–1222, May 2014.

[7] R. Martins, N. Lourenço, N. Horta, J. Yin, P.-I. Mak, and R. P. Martins, "Many-objective sizing optimization of a class-c/d vco for ultralow-power iot and ultralow-phase-noise cellular applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 1, pp. 69–82, Jan. 2019.

[8] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *Proc. 51st Annu. Design Autom. Conf.*, San Francisco, CA, USA, 2014, pp. 1–6.

[9] M. D. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, Jan. 2001.

[10] W. Daems, G. Gielen, and W. Sansen, "An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits," in *Proc. Design Autom. Conf.*, New Orleans, LA, USA, 2002, pp. 431–436.

[11] J. Tao, Y. Su, D. Zhou, X. Zeng, and X. Li, "Graph-constrained sparse performance modeling for analog circuit optimization via SDP relaxation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 8, pp. 1385–1398, Aug. 2019.

[12] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, "An artificial neural network assisted optimization system for analog design space exploration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2640–2653, Oct. 2020.

[13] Y. Yang *et al.*, "Smart-MSP: A self-adaptive multiple starting point optimization approach for analog circuit synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 3, pp. 531–544, Mar. 2018.

[14] A. Garbaya, M. Kotti, N. Drira, M. Fakhfakh, E. Tlelo-Cuautle, and P. Siarry, "An RBF-PSO technique for the rapid optimization of (CMOS) analog circuits," in *Proc. 7th Int. Conf. Mod. Circuits Syst. Technol. (MOCAST)*, Thessaloniki, Greece, 2018, pp. 1–4.

[15] B. Liu, F. V. Fernandez, and G. G. E. Gielen, "Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 6, pp. 793–805, Jun. 2011.

[16] R. A. de Lima Moreto, C. E. Thomaz, and S. P. Gimenez, "Gaussian fitness functions for optimizing analog CMOS integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1620–1632, Oct. 2017.

[17] A. Canelas *et al.*, "FUZYE: A fuzzy c -means analog IC yield optimization using evolutionary-based algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 1, pp. 1–13, Jan. 2020.

[18] M. Barros, J. Guilherme, and N. Horta, "Analog circuits and systems optimization based on evolutionary computation techniques," in *Studies in Computational Intelligence*, vol. 294. New York, NY, USA: Springer, 2010.

[19] T. O. Weber and W. A. M. Van Noije, "Analog design synthesis method using simulated annealing and particle swarm optimization," in *Proc. 24th Symp. Integr. Circuits Syst. Design*, 2011, pp. 85–90.

[20] B. Liu, D. Zhao, P. Reynaert, and G. G. E. Gielen, "GASPAD: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 2, pp. 169–182, Feb. 2014.

[21] A. Ciccazzo, G. Di Pillo, and V. Latorre, "A SVM surrogate model-based method for parametric yield optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 7, pp. 1224–1228, Jul. 2016.

[22] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Conf.*, 2015, pp. 2962–2970.

[23] R. Gómez-Bombarelli *et al.*, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Sci.*, vol. 4, no. 2, pp. 268–276, 2018.

[24] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, "Unscented Bayesian optimization for safe robot grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, 2016, pp. 1967–1972.

[25] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3306–3314.

[26] 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.

[27] N. Knudde *et al.*, "Data-efficient Bayesian optimization with constraints for power amplifier design," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Model. Optim. (NEMO)*, Reykjavik, Iceland, 2018, pp. 1–3.

[28] W. Lyu *et al.*, "An efficient Bayesian optimization approach for automated optimization of analog circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 6, pp. 1954–1967, Jun. 2018.

[29] Y. Bengio, O. Delalleau, and N. Roux, "The curse of highly variable functions for local kernel machines," in *Advances in Neural Information Processing Systems*, vol. 18, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA, USA: MIT Press, 2006, pp. 107–114.

[30] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.

[31] R. Behazd, *Design of Analog CMOS Integrated Circuit*. New York, NY, USA: McGraw-Hill, 2000.

[32] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Conf.*, 2019, pp. 7825–7835.

[33] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Multi-objective Bayesian optimization for analog/rf circuit synthesis," in *Proc. 55th Annu. Design Autom. Conf.*, 2018, pp. 1–8.

[34] 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.

[35] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, Jun. 2009.

[36] N. Lourenço, R. Martins, A. Canelas, R. Póvoa, and N. Horta, "AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation," *Integration*, vol. 55, pp. 316–329, Sep. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167926016300128

[37] *WiCkeD User Manual*, 6.7 ed., MunEDA, Unterhaching, Germany, 2014.

[38] E. Bradford, A. M. Schweidtmann, and A. Lapkin, "Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm," *J. Global Optim.*, vol. 71, no. 2, pp. 407–438, 2018.

[39] G. Pleiss, J. Gardner, K. Weinberger, and A. G. Wilson, "Constant-time predictive distributions for Gaussian processes," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4114–4123.

[40] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Conf.*, 2019, pp. 5497–5508.

[41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[42] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Poczos, "Parallelised Bayesian optimisation via Thompson sampling," in *Proc. 21st Int. Conf. Artif. Intell. Stat.*, vol. 84, Apr. 2018, pp. 133–142.

[43] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2014, pp. 918–926.

[44] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, vol. 20. Cambridge, MA, USA: MIT Press, 2007, pp. 1177–1184.

[45] J. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth, "Efficiently sampling functions from Gaussian process posteriors," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, Jul. 2020, pp. 10292–10302.

[46] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 1–20, Feb. 2021.

[47] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 7576–7586.

[48] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[49] Z. Yan, P.-I. Mak, M.-K. Law, and R. Martins, "A 0.016 mm^2 144 μW three-stage amplifier capable of driving 1-to-15nF capacitive load with >0.95 MHz GBW," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2012, pp. 368–370.

[50] S. A. Fordjour, J. Riad, and E. Sánchez-Sinencio, "A 175.2-mW 4-stage OTA with wide load range (400 pF–12 nF) using active parallel compensation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 7, pp. 1621–1629, Jul. 2020.

[51] E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes," *Neurocomputing*, vol. 380, pp. 20–35, Mar. 2020.

[52] L. Mendes, J. C. Vaz, F. Passos, N. Lourenço, and R. Martins, "In-depth design space exploration of 26.5-to-29.5-GHz 65-nm CMOS low-noise amplifiers for low-footprint-and-power 5G communications using one-and- two -step design optimization," *IEEE Access*, vol. 9, pp. 70353–70368, 2021.

Konstantinos Touloupas (Student Member, IEEE) received the M.S. degree in electrical and computer engineering from the National Technical University of Athens, Zografou, Greece, in 2017.

He has authored IEEE conferences. His major research area and interests include machine learning, optimization, and their application in electrical and electronic engineering.

Mr. Touloupas is a regular reviewer for many IEEE transactions and conferences.

Paul P. Sotiriadis (Senior Member, IEEE) received the Diploma degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 1994, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1996, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2002.

In 2012, he joined the faculty of the Electrical and Computer Engineering Department, National Technical University of Athens, Zografou, Greece, where he is a Faculty Member and the Director of the Electronics Laboratory. He is a member of the governing board of the Hellenic Space Center, the National space center of Greece. In 2002, he joined the Johns Hopkins University, Baltimore, MD, USA, as an Assistant Professor of Electrical and Computer Engineering. He has authored and coauthored more than 150 research publications, most of them in IEEE journals and conferences, holds one patent, and has contributed chapters to technical books. He has led several projects in these fields funded by U.S. organizations and has collaborations with industry and national labs. His research interests include design, optimization, and mathematical modeling of analog and mixed-signal circuits, RF and microwave circuits, advanced frequency synthesis, biomedical instrumentation, and interconnect networks in deep-submicrometer technologies.

Dr. Sotiriadis has received several awards, including the 2012 Guillemin-Cauer Award from the IEEE Circuits and Systems Society, the Best Paper Award in the IEEE International Symposium on Circuits and Systems 2007, the Best Paper Award in the IEEE International Frequency Control Symposium 2012, and the Best Paper Award in the IEEE International Conference on Modern Circuits and Systems Technologies 2019. He is an Associate Editor of the IEEE SENSORS JOURNAL, has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS from 2016 to 2020 and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS from 2005 to 2010, and has been a member of technical committees of many conferences. He regularly reviews for many IEEE transactions and conferences and serves on proposal review panels.