

An Algorithm for Exploiting Modeling Error Statistics to Enable Robust Analog Optimization

Ashish Kumar Singh, Mario Lok, Kareem Ragab, Constantine Caramanis, and Michael Orshansky

Department of Electrical and Computer Engineering

University of Texas at Austin

Email: aksingh@cerc.utexas.edu

Abstract—Equation-based optimization using geometric programming (GP) for automated synthesis of analog circuits has recently gained broader adoption. A major outstanding challenge is the inaccuracy resulting from fitting the complex behavior of scaled transistors to posynomial functions. Fitting over a large region can be grossly inaccurate, and in fact, poor posynomial fit can lead to failure to find a true feasible solution. On the other hand, fitting over smaller regions and then selecting the best region, incurs exponential complexity.

In this paper, we advance a novel optimization strategy that circumvents these dueling problems in the following manner: by explicitly handling the error of the model in the course of optimization, we find a potentially suboptimal, but feasible solution. This solution subsequently guides a range-refinement process of our transistor models, allowing us to reduce the range of operating conditions and dimensions, and hence obtain far more accurate GP models.

The key contribution is in using the available oracle (SPICE simulations) to identify solutions that are feasible with respect to the accurate behavior rather than the fitted model. The key innovation is the explicit link between the fitting error statistics and the rate of the error uncertainty set increase, which we use in a robust optimization formulation to find feasible solutions.

We demonstrate the effectiveness of our algorithm on a two benchmarks: a two-stage CMOS operational amplifier and a voltage controlled oscillator designed in TSMC 0.18 μ m CMOS technology. Our algorithm is able to identify superior solution points producing uniformly better power and area values under gain constraint with improvements of up to 50% in power and 10% in area for the amplifier design. We also demonstrate that when utilizing the models with the same level of modeling error, our method yields solutions that meet the constraints while the violations for the standard method were as high as 45% and larger than 15% for several constraints.

I. INTRODUCTION

One of the challenging aspects of analog design is optimizing a given circuit topology to meet design specifications, such as gain, while minimizing cost metrics such as area and power. This process poses severe challenges due to the stringent requirements upon multiple mutually conflicting performance constraints. In any manual design strategy, success heavily depends on the designer's experience and design-specific intuition. Automated analog optimization promises to increase productivity by reducing design time. Efforts to automate analog design have taken two major routes. In one, the circuit topology is assumed to be fixed and only optimal device sizing is performed [8], [10]. In the other, the topology

is also selected automatically [11]. Our work focuses on the first class of approaches.

The existing work has fallen into two major categories based on how they evaluate a solution point to drive optimization: approaches relying on extensive use of SPICE simulations [11], [12], [9], and those that construct an analytical model of circuit behavior and use the model to drive the optimization [5], [10], [13], [6]. Clearly, evaluating a solution point via a SPICE simulation gives the most accurate measure of the feasibility and optimality of a solution. However, SPICE simulations are time-consuming, and yield no structure that can be exploited for efficient global optimization. Both local and global solution search methods relying on SPICE simulations have been studied. The local search methods typically are based on gradient methods, while global solution methods employ simulated annealing or genetic algorithms. In local search methods performance is tied to the initial points of the algorithm. Global search methods require a potentially much larger number of SPICE evaluations for the global exploration to converge [12], and typically have exponential complexity.

The model-based, or equation-based methods help to avoid the circuit-level SPICE simulation costs. They rely on a model of circuit behavior to capture the constraints and the objective function in terms of design variables. The typical optimization variables are transistor width, length, biasing currents, and the constraints are bounds on gain, bandwidth, and other performance measures. The models express in closed form the circuit performances in terms of the small-signal transistor parameters such as transconductance (g_m) and output conductance (g_d), transistor capacitances, overdrive voltages, sizes, currents etc.

The challenge of model-based optimization is two-fold: (1) to ensure sufficient model accuracy, or at least, fidelity, and (2) do that in the functional form that lends itself to efficient optimization. The first-principles small-signal parameters derived based on long-channel transistor theory are not accurate for nanometer scale technologies. The small signal parameters g_m , g_d , g_{mb} , as well as overdrive voltage and the transistor capacitances, are complex functions of the biasing current and sizes. Device-level modeling of transistors is embedded into SPICE device models, such as BSIM4, and involves hundreds of variables. Directly working with such models appears infeasible since in order to be tractable equation-based optimization requires continuous low-dimensional mod-

els, and, in particular, convexity.

In fitting regression models we need to utilize functional forms allowing efficient convex optimization. Specifically, it has been observed that many of the circuit performance constraints can be cast as a special class known as *posynomial* functions [5]. Optimization problems using posynomial constraints and objective function can be solved efficiently using the convex optimization framework of the geometric programming (GP) paradigm relying on fast interior point solution methods. Posynomials are generalized polynomials with positive coefficients and arbitrary real exponents [14]. Thus, a posynomial $f(x_1, \dots, x_n)$ has the general form $\sum_{q=1}^{\beta} a_q x_1^{\alpha_{q1}} \dots x_n^{\alpha_{qn}}$ where the α 's are arbitrary, and the a 's are positive. Each product term is called a *monomial*.

Relying on posynomial models is useful due to the fact that they yield globally-solvable tractable optimization problems. The cost, however, is that the fitting error of posynomial models can be large. While in part, this is due to the complex device physics, another important source of error is due to the limitations imposed by the requirement of modeling in a GP-compatible manner. In addition to limiting our fitting capability, GP-compatibility forces us to eliminate referencing some key input variables, e.g., it has been proven difficult to capture the dependence on V_{ds} (drain to source voltage) in posynomial form, in contrast to dependence on I (drain current), W (width), and L (gate length) [6]. As a result, the optimization model may differ greatly from the behavior of the physical device. As we quantitatively demonstrate later, the attributes of the solution generated by the optimization may be more than 30% off from the intended target constraints.

The inaccuracy of posynomial fitting over a large range has been observed in the literature [8], [10]. In [8], the proposal to model transistor parameter behavior using piecewise linear models improves accuracy, but may lead to non-posynomial constraints, hence rendering convex optimization unusable. Elsewhere in the literature, the modeling error is either fully ignored [6], or local refinement methods are used [10], or generic robustification strategies are adopted [8] that robustify the optimization, but, crucially, not in a way directly linked to the errors in the specific model's fit. The work in [10] does local search in the small vicinity of the current solution. In comparison, single transistor fitting which requires three or four variables, can be done over a broader range with relatively better accuracy. In [8], a simple strategy for robustification utilizes the worst-case error. This can be highly over-conservative because it may introduce robustness where none is needed. As a consequence, the feasible space of the optimization may be unduly and significantly reduced, resulting in degraded performance, or in the worst case, an inability to find a SPICE-simulation feasible (we henceforth call this *true-feasible*) solution.

Main Contributions and Outline: The power of geometric programming (GP) lies in its ability to quickly and efficiently find guaranteed optimal solutions to the model represented by the given posynomials. The fundamental shortcoming is the inherent inability to accurately model the behavior of transistor

parameters over a large range of values of width, length, and biasing currents. The central contribution of this paper is to harness the power of the convex optimization approach, while mitigating its significant drawbacks, thus moving us closer towards automating analog design and optimization. More specifically:

- (1) We develop an efficient iterative algorithm that converges to a good solution that meets multiple performance constraint targets. This is in sharp contrast to existing algorithms that may be off by upwards of 30% from the desired targets.
- (2) We develop the notion of the *coverage metric* that allows us to map fitting error to robust optimization uncertainty sets. We show that this coverage metric serves as a successful proxy for true feasibility (i.e., SPICE-verified feasibility).
- (3) Using this coverage metric, we develop a principled robust optimization-based method that explicitly incorporates fitting error into the circuit optimization, allowing us to optimize objectives while also meeting performance constraints exactly. We show that the coverage metric can be optimized via a linear or binary search to reduce the conservatism in uncertainty set and still obtain a true feasible solution.
- (4) Our method is independent of the fitting procedure used, and hence is flexible and modular. Thus, we believe the tools introduced here could be important for a broader class of problems.

The remainder of the paper is organized as follows. In Section II, we lay out the conceptual framework of our main approach. In Section III, we provide the details of the optimization. Finally, in Section IV, we provide the numerical experiments that show the performance of our algorithm.

II. EXPLOITING MODELING ERROR STATISTICS TO DRIVE OPTIMIZATION

A. High-level Algorithm and Approach

Our strategy is based on two key observations. First, while any given posynomial model may have significant errors, we can precisely assess the true feasibility of the solution using SPICE. Thus, despite the difficulty in obtaining a globally accurate fit, we have a local oracle of true feasibility. However, while we can determine true feasibility of any given point $(\mathbf{W}, \mathbf{L}, \mathbf{I})$, i.e., membership in the set of true feasible points, we cannot optimize over this set — indeed this problem is known to be NP-hard and hence intractable. Second, is the observation that the fitting error is a product of the global fit. Obtaining a better fit over a smaller range is, not surprisingly, much more readily achievable. Yet a brute-force search for a “good” limited range of variables (width, length, etc) over which to fit the transistor parameters and, subsequently, to optimize, is a hopeless avenue, as its computational complexity grows exponentially. Even seeking to reduce the range of each variable by just a factor of $1/d$, would lead to d^N possible variable ranges to consider, where N is the number of modeled transistor parameters.

A central part of our contribution is a robust optimization-based scheme for efficiently finding such a “good” limited range for the variables, over which the quality of the posynomial fit to the true behavior of the transistor parameters is significantly improved. This refinement relies on explicitly modeling the error of the posynomial fit and incorporating this through robust optimization in the circuit optimization.

We denote a solution by $(\mathbf{W}, \mathbf{L}, \mathbf{I})$, where bold-face indicates vector notation, i.e., \mathbf{W} , \mathbf{L} and \mathbf{I} are the vectors of all the width, length and current variables respectively in the circuit.

Most *standard* equation-based optimization flows that have been proposed rely on a sequence of basic steps in setting up the optimization [6], [8], [10]. First, the simulation data generated by SPICE are used to fit the parameters of each transistor, using linear least-squares regression. Second, the fitted models are used within equations that capture the design specifications, such as gain, bandwidth. Finally, the resulting geometric program is solved using standard or specialized convex optimization solvers (e.g., [4], [1]) to obtain $(\mathbf{W}, \mathbf{L}, \mathbf{I})$.

On top of standard GP-based methods, our algorithm adds two enhancements: robustness to fitting error, and range refinement. Adding robustness aims to ensure that we find solutions that SPICE simulations show to be feasible to the design constraints, an integral part of our refinement strategy. The distinction between model and SPICE-feasibility is a crucial operating principle in our scheme. We refer to variable values $(\mathbf{W}, \mathbf{L}, \mathbf{I})$ as *model feasible* if they are feasible for the optimization problem at hand, and call them *true feasible* if SPICE simulations show that they meet the target constraints. We add robustness to fitting error using the paradigm of robust optimization (see [2] for details). The essence of robust optimization is to build in deterministic protection to parameter uncertainty in a chosen uncertainty set \mathcal{U} . That is, given \mathcal{U} , the solution to the resulting robust optimization problem is *guaranteed to be feasible* under any variation of the optimization parameters in the given uncertainty set.

The nominal optimization problem is a special case of a robust optimization problem, where there is no uncertainty, i.e., where the uncertainty set \mathcal{U} is empty. The regression procedure fits posynomials to the individual parameters of each transistor. The key idea of our algorithm is that *the fitting error of regression fit can be treated as the parameter-error in the robust optimization setting*. Thus, we have an uncertain variable for every fitted function. As an example, if we have a design constraint that has the form:

$$\frac{g_m(W, L, I)}{g_d(W, L, I)} \geq c,$$

the robust version of this would then become:

$$\frac{g_m(W, L, I) \cdot e_1}{g_d(W, L, I) \cdot e_2} \geq c, \quad \forall (e_1, e_2) \in \mathcal{U},$$

meaning that the constraint must be satisfied for all values of the error parameters e_1 and e_2 in the uncertainty set \mathcal{U} . Using duality techniques from convex optimization, we reformulate these constraints in a tractably solvable fashion (see [7], [2] for details).

The existing theory of robust optimization does not tell us how the uncertainty sets should be chosen. Addressing this issue is an important part of our contribution. As we discuss in detail in Section II-B, the uncertainty sets we design are chosen in an optimal manner, in order to account for the fitting errors. We use robustness as a tool to guarantee that the solution is *true-feasible*.

The high-level description of our algorithm is as follows.

- (1) **Initialize:** Apply the standard procedures to obtain $(\mathbf{W}_0, \mathbf{L}_0, \mathbf{I}_0)$ ¹.
- (2) **Find Feasible Solution:** If the solution $(\mathbf{W}, \mathbf{L}, \mathbf{I})$ is not *true feasible*, then we increase the robustness of our algorithm to the fitting errors, by solving a robust GP over a larger uncertainty set, \mathcal{U} . We repeat this step, increasing the size of the uncertainty set until we find a true-feasible solution, $(\mathbf{W}_f, \mathbf{L}_f, \mathbf{I}_f)$. The algorithm for designing and increasing the uncertainty set \mathcal{U} is described below, in Section II-B.
- (3) **Refine Variable Range:** Given the feasible point $(\mathbf{W}_f, \mathbf{L}_f, \mathbf{I}_f)$, we refine the allowed range of the variables, and return to step (1), performing a new (and hence better) posynomial fit to the transistor parameters. At each step, we refine the range by shrinking it by a constant multiplicative factor.

B. Robustness and True Feasibility

Step 2 in our high-level algorithm description above, requires us to enlarge the uncertainty set each time the optimization outputs a solution that is not true-feasible. Thus, the problem faced is to find the “optimal” uncertainty set: the uncertainty set that yields a true-feasible solution, while attaining the best possible objective value. In the case of robust GP, we seek uncertainty sets \mathcal{U} that are either ellipsoidal or rectangular. Even for this simpler class of sets, computing the optimal set can be shown to be nonconvex and intractable, and certainly a brute-force search would require effort exponential in the dimension.

What Does Not Work. It is helpful to discuss two potential naive approaches. One approach is to simply let \mathcal{U} be a uniform box of dimension equal to the number of transistor parameters, and then slowly increase its size, hoping that a true-feasible solution is found before the problem becomes infeasible. This reduces the search to a single dimension (since the dimensions of the box are scaled in lock step) and hence is tractable. While this strategy does sometimes succeed in obtaining a true-feasible solution, we have found that it also often leads to model-infeasibility, and in either case, the objective value suffers more than required. The intuition as to why such an approach fails, is simple: we may be adding too much robustness where none is needed, and too little where more is needed. Another possible approach is to solve the nominal problem, look at constraints that are violated according to SPICE, and increase the robustness requirement of the transistor parameters participating in those

¹That is, fitting parameters over a broad range, and solving the GP.

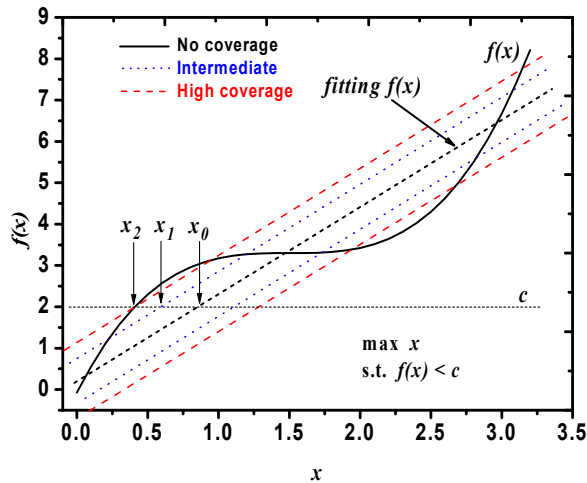


Fig. 1. The growth of uncertainty set based on coverage increase is demonstrated in the above one dimensional optimization problem.

true-infeasible constraints. The conceptual, and as we have found, practical problems with such an approach, are that we are basing our uncertainty set selection (and ultimately our iterative optimization strategy) on the behavior of an infeasible point.

What Does Work. We formulate an uncertainty-set selection approach that tackles the fundamental fitting problem head-on: we design our uncertainty sets based on the error in the fit for each parameter. In order to do this, we introduce the idea of coverage. Consider some parameter of a given transistor – for concreteness suppose it is the transconductance parameter g_m . Recall that each such parameter corresponds to an uncertain parameter, in this case let us refer to it as e_1 . The coverage captures how much of the true function behavior, as represented by SPICE-generated tables, is covered by the fitted function with an added uncertainty set. Formally, given an uncertainty set \mathcal{U} , we define the *coverage metric of parameter g_m* to be the fraction of entries in the SPICE-generated tables of transistor behavior for which the exact value of the parameter is equal to the value given by the fitted function, g_m , times some error $e_1 \in \mathcal{U}$. Thus, when \mathcal{U} is empty, the coverage is the fraction of points in the table for which the fitted function is exact. As the uncertainty set \mathcal{U} grows, the coverage increases to 100% (Figure 1). Note that this metric captures the error of the posynomial fit, and in particular has the following two important properties: (a) it is not sensitive to outliers, since it simply computes the fraction of points whose fit is within the tolerance of the uncertainty set – if a point is not in that set, it is simply not counted, and how far outside it is does not matter; and (b) it is a *global metric* and does not depend on a true-infeasible point of the range.

Given two transistor parameters, we can define their joint coverage to be the minimum of their coverages. Similarly, we can now define the overall coverage as the infimum of all the individual coverages. This gives us a meaningful measure for

the size of the uncertainty set. It should be noted that the “size” is not *a priori* well-defined in the context of a general robust optimization problem, e.g., something like volume may well be irrelevant in the context of robust optimization. Coverage allows us to compare two arbitrary uncertainty sets, and hence coverage becomes a proxy for how strongly a set pushes the optimization problem to produce a true-feasible point.

We now present a formal high-level description of our algorithm for increasing the uncertainty sets. We do this by solving a bi-criterion problem which has the form of a geometric program. The key observation is that maximizing coverage subject to the constraint that the objective value of the problem corresponding to the uncertainty set is no more than some fixed value α , can be rewritten as a convex optimization problem. Thus, the high-level algorithm for Step 2 of the algorithm given above, finding a feasible solution, now becomes:

- (i) If the problem from Step (2) is true-infeasible, increase the coverage requirement by the chosen step-size, Δ : $p \leftarrow p + \Delta$.
- (ii) Find the minimum value of α such that maximizing the coverage subject to the constraint that the objective value of the corresponding robust geometric program is at most α , gives coverage exceeding the chosen value, p . We solve the search over α by bisection since the problem is quasi-convex.
- (iii) With this uncertainty set, solve the robust geometric program. If the returned solution is true feasible, then move on to Step (3) of the main algorithm. If it is not true-feasible, then increase the coverage requirement by Δ , and return to step (ii).

III. FORMAL DESCRIPTION OF THE ALGORITHM

We now formally describe the proposed analog circuit optimization scheme. First, we require the fitted model for each transistor parameter needed to set up the constraints and the objective which are in terms of various performance metrics such as bandwidth, loop-gain, power. A set of SPICE simulations is carried out at the characterization phase to create a table capturing the values of the channel conductance (g_d) and transconductance (g_m), overdrive voltage (V_{gt}), and transistor capacitances (C_{gs} , C_{gd} , and C_{gb}). These measurements are made over a range of transistor width, length and bias current values, separately for NMOS and PMOS transistors.

Next, a model is fitted to the collected data. We perform a single monomial (unknown exponents) fit by applying a log transformation and subsequently taking the exponential to recover the transistor parameters as a single monomial. Via regression, we obtain coefficients and exponents for the best-possible, in the least square sense, monomial fit. Thus, for example, our model for g_m becomes: $g_m \approx aW^bL^cI^d$, where values a , b , c and d are determined through regression to the data in the table.

Now let there be q transistor parameters $M = (m_1, \dots, m_q)$, where each m_i is a fitted monomial function of (W, L, I) . The nominal constraints of the circuit optimization

are built up from the $\{m_i\}$ in a manner consistent with geometric programming, namely, in a multiplicative manner. Thus, constraints will take the form:

$$\frac{\prod_{i \in I} m_i}{\sum_{l \in L} \prod_{j \in J_l} m_j} \geq 1.$$

We can now describe the robust optimization problem. Robust GP is tractable for interval and ellipsoidal uncertainty sets, and we use the former. To be consistent with the multiplicative nature of geometric programming, we use a multiplicative model for the error with an error parameter e_i corresponding to each transistor parameter m_i . In the interval uncertainty model, each e_i is constrained to lie in an interval $[-k_i, k_i]$. (For example, for the transconductance parameter g_m , we have $g_m = aW^b L^c I^d \exp(e)$, where the error term e belongs to an interval $[-k, k]$.)

For multiple constraints, we express the uncertainty set as

$$\mathcal{U} \triangleq \mathcal{U}(\mathbf{k}) = \prod_{i=1}^q [-k_i, k_i],$$

which is thus parameterized by the vector $\mathbf{k} = (k_1, \dots, k_q)$. The *robustified constraints* now have the form:

$$\frac{\prod_{i \in I} m_i \cdot \exp e_i}{\sum_{l \in L} \prod_{j \in J_l} m_j \cdot \exp e_j} \geq 1, \quad \forall (e_1, \dots, e_q) \in \mathcal{U}(\mathbf{k}).$$

For a fixed uncertainty set $\mathcal{U}(\mathbf{k})$, letting M denote the set of transistor parameters as above, and V denote the complete set of variables (W, L, I) for each transistor, the robust GP that we need to solve is:

$$\begin{aligned} \text{Min}_V : & \quad \text{Objective}(V) \\ \text{s.t.} : & \quad \text{Constraints}(V, M, \mathbf{k}) \leq 1 \end{aligned} \quad (1)$$

The objective function, like the constraints, is a posynomial function of transistor parameters and variables in V .

Selecting Coverage and Robustness: In the first iteration, we set $\mathcal{U}(\mathbf{k}) = \{\mathbf{0}\}$, i.e., we have $\mathbf{k} = \mathbf{0}$. Each time the robust GP returns a solution that is not feasible, we must increase the uncertainty set \mathcal{U} , by increasing the vector \mathbf{k} . Note that this cannot be done by sweeping or by brute force, since any such attempt is exponential in q , the number of transistor parameters.

Recall from Section II, that given an uncertainty set \mathcal{U} , the *coverage* of a transistor parameter, m_i , is equal to the fraction of points in the table (in the range we are considering) where the error falls within \mathcal{U} . Formally, given $\mathcal{U}(\mathbf{k})$, the coverage of parameter m_i , which we denote as $\text{Coverage}(k_i, i)$, is equal to the number of entries in the table where the error is within a multiplicative factor of $\exp(\pm k_i)$. The overall coverage, which we denote by $\text{Coverage}(\mathbf{k})$, is then just the minimal of the local coverages: $\text{Coverage}(\mathbf{k}) = \text{Min}_{i=1}^q \text{Coverage}(k_i, i)$. Our algorithm calls for us to maximize the overall coverage, while decreasing the objective of the resulting robust optimization problem as little as possible. We expand the uncertainty set optimally, by controlling the allowable decrease in optimality of the objective function by a factor of $\alpha\%$. Thus, we select

the optimal \mathbf{k} by solving the now-posynomially-expressed optimization problem

$$\begin{aligned} \text{Maximize}_{\mathbf{k}, V} & \quad \text{Coverage}(\mathbf{k}) \\ \text{s.t.} : & \quad \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{\alpha}{100}) \\ & \quad \text{Constraints}(V, M, \mathbf{k}) \leq 1 \end{aligned} \quad (2)$$

In the above problem the constraint set is a collection of posynomials, however, the objective is not. We found that it cannot be well approximated by a monomial expression needed for GP compatibility. We also do not have the option to use a posynomial fit with multiple terms since this will violate GP compatibility. The objective function being maximized must be a monomial. As an alternative, we use a bisection-based approach, which we describe below.

We note that coverage value is between 0 and 1. In order to enable the bisection approach, we need to verify whether there exists a solution in the above constraint space for which coverage is greater than a number $\beta \in [0, 1]$. If we can verify this, then it is clear that a bisection approach will help us find optimal coverage. Thus, we focus on describing the procedure to check if $\text{Coverage}(\mathbf{k}) \geq \beta$. Note that this is equivalent to saying that the coverage of each individual parameter is greater than β , i.e. $\text{Coverage}(k_i, i) \geq \beta$. Letting F_i denote the empirical distribution of the absolute value of the fitting error for the log of parameter m_i , we can deduce equivalently that $F_i(k_i) \geq \beta$. We treat $\exp(k_i)$ as a problem variable k'_i and denote the vector $\exp(\mathbf{k})$ as \mathbf{k}' . By taking the inverse of the distribution function, we get $k'_i = \exp(k_i) \geq \exp(F_i^{-1}(\beta))$. Now in order to check the feasibility, we check the feasibility of the following problem, which is a geometric program in terms of the variables V and \mathbf{k}' :

$$\begin{aligned} \text{Maximize}_{\mathbf{k}', V} & \quad 1 \\ \text{s.t.} : & \quad \text{Objective}(V) \leq \text{obj}_i * (1 + \frac{\alpha}{100}) \\ & \quad \text{Constraints}(V, M, \mathbf{k}') \leq 1 \\ & \quad \mathbf{k}'_i \geq \exp(F_i^{-1}(\beta)) \forall i \in [1, q] \end{aligned} \quad (3)$$

Thus, each time the optimization problem returns a solution that is not true feasible, we seek the uncertainty set $\mathcal{U}(\mathbf{k})$ that maximizes coverage, while not increasing the objective function by more than $\alpha\%$ of the current objective obj_i .

We can now describe all the steps of the algorithm as follows.

Input: SPICE-table, initial range, user-selected parameter Δ . Initialize $\alpha = 0$ and $\mathbf{k} = \mathbf{0}$.

- 1) Solve Problem (1) to obtain solution $(\mathbf{W}, \mathbf{L}, \mathbf{I})$.
- 2) While $(\mathbf{W}, \mathbf{L}, \mathbf{I})$ is not true feasible, set $\alpha \leftarrow \alpha + \Delta$. Solve Problem (2) to obtain the new vector \mathbf{k} and then solve Problem (1) to obtain new solution $(\mathbf{W}, \mathbf{L}, \mathbf{I})$.
- 3) While the range is larger than the minimum size, shrink the range around the true-feasible solution $(\mathbf{W}, \mathbf{L}, \mathbf{I})$, set $\alpha = 0$ and $\mathbf{k} = \mathbf{0}$, and return to Step 1.
- 4) Report true-feasible solution $(\mathbf{W}, \mathbf{L}, \mathbf{I})$.

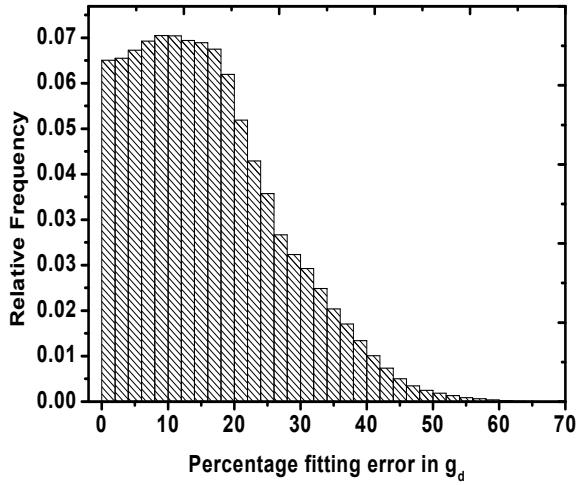


Fig. 2. The histogram shows the distribution of fitting errors for g_d for a PMOS transistor.

IV. EXPERIMENTAL RESULTS

In this section, we report our numerical experiments to validate the algorithm performance. Devices were characterized using 180nm TSMC high-performance technology models. First, we illustrate the high fitting errors in the monomial equations needed for GP. In Figure 2, we show the histogram of the fitting errors for the output conductance parameter g_d of a PMOS transistor. The transistor is simulated in HSPICE to predict the g_d and drain current for a set of width, length, V_{gs} and V_{ds} values. The samples are generated by varying the gate length from 180nm to $1.8\mu\text{m}$ and gate width varying from 180nm to $18\mu\text{m}$, both in increments of 20%. The V_{gs} ranges from 0.65 to 1.8V and V_{ds} ranges from 0.35 to 1.8V. Then g_d was fitted as a monomial function of width, length, and drain current.

The fitted equation was $g_d = 0.079W^{0.22}L^{-0.84}I^{0.73}$, where the unit of g_d is $\frac{\mu\text{A}}{\text{V}}$, W and L are in units of μm , and I is measured in unit of μA . The fitting errors are significant: Figure 2 shows that a high number of samples yielded a fitting error higher than 20%. The rms error of the fit is 19%. Importantly, the maximum error is 69%. This indicates that there is a danger of optimizing around a region in which the model fit is very poor. Similar trends are observed for other fitted functions.

We next demonstrate the improvement in fitting accuracy through refinement by an example shown in Figure 3. To simplify presentation, we restrict the fit to a one-dimensional single variable fit. We do refinement by generating samples with W , L and V_{ds} fixed, and varying only V_{gs} from 0.65 to 1.8V in increments of 0.05. We show the fitting of g_d for a PMOS transistor as a monomial function of current. The worst-case error is 10% when fitting over the V_{gs} range of 0.65 to 1.8V. However, when we restrict the range of V_{gs} to be 1.05 to 1.8V, the worst-case error is reduced to 1.3%. The point is that even modest refinement of the range (factor of 2 here) can dramatically improve the fitting error (about a factor

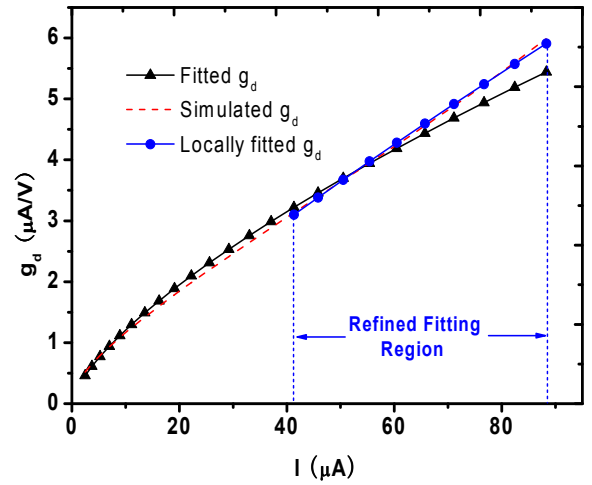


Fig. 3. Fitting improvement with refinement of the fitting region. The errors of the global fit are as large as 10%. The worst-case error is 1% for the narrower region.

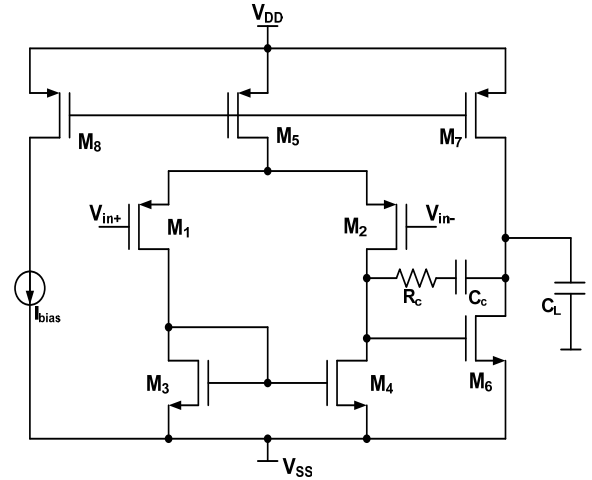


Fig. 4. The two-stage operational amplifier used for numerical experiments.

of 8, here).

We next report the outcomes of numerical experiments that validate the performance of our algorithm, and, in particular, compare our algorithm to the existing global GP-based solution. We compare the proposed *robustness and refinement* (RAR)-based optimization with the prior equation-based global search method employing geometric programming. We refer to the prior method as the standard optimization. We demonstrate the effectiveness of our algorithm by using it to optimize the area of a two-stage CMOS operational amplifier and power for a voltage controlled oscillator, which have been used as validation vehicles in several prior related publications [10], [3] (Figure 4 and Figure 5). The two-stage amplifier circuit is made up of 8 transistors and the typical design metrics include gain, unity gain bandwidth, slew rate, common-mode rejection ratio, phase margin, as well as area. The voltage controlled oscillator has min and max frequency constraints as well as saturation constraints for all the transistors. We also have a

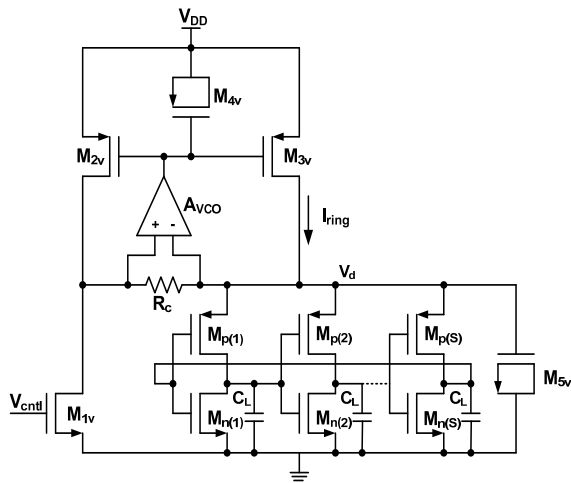


Fig. 5. The voltage controlled oscillator used for numerical experiments.

constraint on transistor sizes which sets the transistor lengths and widths within the range of 180nm to 1.8 μ m. Our algorithm refines the variable range by 20% in each iteration, after finding a true feasible solution. We set 10 as the maximum number of refinement steps.

The op-amp design process is intrinsically a multi-objective optimization process and the optimal solutions lie on the multidimensional Pareto-surfaces. It is hard to present Pareto curves in more than two dimensions, so we first demonstrate the effectiveness of the algorithm by showing the value of the objective function that can be obtained for a single constraint. We show the value of amplifier gain (used as the constraint) against the objective area, in one case, and power, in the other case. In the first experiment the Pareto-curve was generated by sweeping the value of target gain over the range of [74.3db 75.3db] and optimizing the power using the standard optimization method. The results are shown in Figure 6 and indicate that we can obtain uniformly better solutions with up to 50% savings in power. We generated a similar tradeoff curve for minimum area in a different range of gain values. This experiment also demonstrates that our algorithm produces uniformly better solutions with up to 10% area savings. The results are shown in Figure 7.

A major benefit of the proposed algorithm is that it offers a guaranteed way of meeting multiple design specifications. Thus, the second set of experiments on solving multiple-constraint problems aims at demonstrating the degree to which the standard method can be infeasible, while our method meets all of the constraints. We find that because of the large fitting errors the standard optimization method often produces solutions that grossly violate the constraints, especially when multiple constraints are used.

In the experiment for the two-stage amplifier, we use area and power (individually) as the objective to minimize, and have constraints on gain, unity gain bandwidth (UGB), slew rate, common-mode rejection ratio (CMRR), phase margin (PM), and negative power supply rejection ratio (PSRR). In

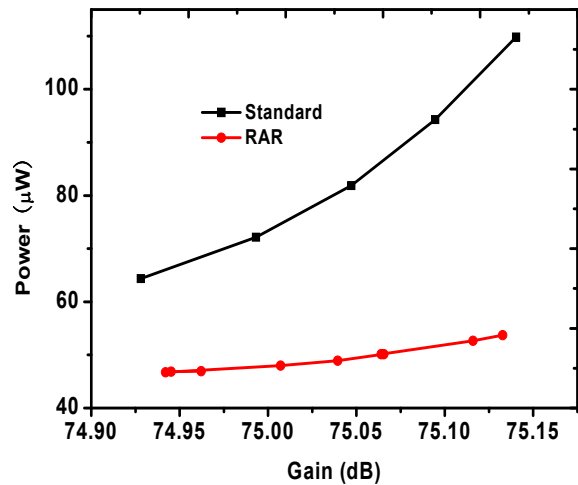


Fig. 6. Power vs. gain Pareto curve. Proposed algorithm is uniformly better and maximum power savings are 50% at fixed gain.

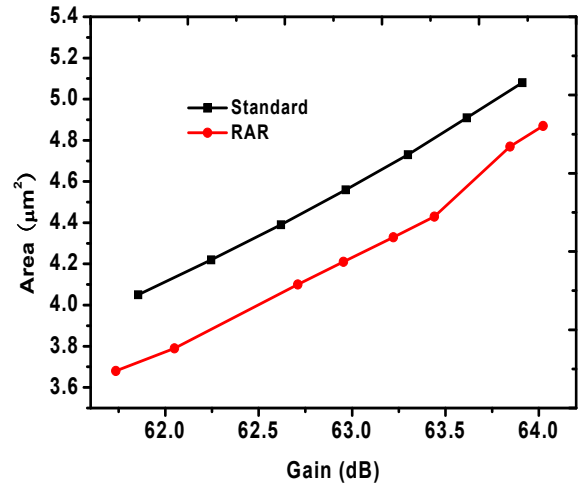


Fig. 7. Area vs. gain Pareto curve. Proposed algorithm is uniformly better and maximum area savings are 10% at fixed gain.

Table I, we present the comparison results in terms of percentage of constraint violations for minimum area optimization. As the results demonstrate, while our algorithm meets the target constraints, the standard optimization is unable to find a feasible solution, and the solution produced in some cases grossly violates the target constraints. We show similar results for a voltage controlled oscillator in Table II.

We note that the objective function of the standard approach appears better than what our approach finds – however, given that the standard optimization produces solutions that violate constraints by up to 47%, it is not clear that the objective function value is meaningful, or even how to devise a fair numerical comparison. A useful comparison would be based on comparing Pareto surfaces, generated by sweeping the values of all the constraints and plotting against the optimal values obtained, which is difficult to do for problems with multiple constraints.

Performance	Spec	RAR	Standard	Spec	RAR	Standard	Spec	RAR	Standard
		% violation	% violation		% violation	% violation		% violation	% violation
Gain (dB)	≥ 66	0	16	≥ 67.3	0	18.5	≥ 65.6	0	15.1
UGB (MHz)	≥ 5	0	0	≥ 5.5	0	0	≥ 6	0	0
Slew Rate (V/ μ s)	≥ 9	0	1.7	≥ 10	0	0	≥ 5	0	0
CMRR (dB)	≥ 66	0	47.8	≥ 65.9	0	31.7	≥ 64.6	0	44.2
Phase Margin (deg)	≥ 60	0	0	≥ 45	0	0	≥ 60	0	0
Negative PSRR (dB)	≥ 74.8	0	17.8	≥ 74.8	0	12.8	≥ 74	0	11.4
Area (μm^2)	MIN	329	237	MIN	296.4	219	MIN	270.4	228.8

TABLE I

Area minimization for the two-stage amplifier benchmark circuit. The standard method leads to significant constraint violations while the proposed method is able to meet all the constraints.

Performance	Spec	RAR	Standard	Spec	RAR	Standard	Spec	RAR	Standard
		% violation	% violation		% violation	% violation		% violation	% violation
Min VCO freq. (GHz)	≤ 1	0	30	≤ 1.2	0	28.3	≤ 1.2	0	30.8
Max VCO freq. (GHz)	≥ 1.25	0	0	≥ 1.4	0	0	≥ 1.5	0	0
Power (μ W)	MIN	69.9	36.7	MIN	69.4	44.6	≥ 5	76.2	50.13

TABLE II

Power minimization in a voltage controlled oscillator benchmark circuit. The standard method leads to significant constraint violations while the proposed method is able to meet all the constraints.

V. CONCLUSION

In this paper we presented a set of algorithmic solutions that aim to explicitly utilize the knowledge of modeling error in the fitted equations to drive optimization. The algorithm is based on two key concepts of refinement and robustness. A novel concept of coverage is used to optimally construct the uncertainty sets. The results are promising and show that significant improvements are possible in terms of the value of the achievable cost functions as well as in terms of reliably meeting performance constraints in the presence of large modeling error.

ACKNOWLEDGMENT

The authors would like to thank Ku He for help with some experiments and illustrations.

REFERENCES

- [1] Mosek ApS, Mosek (Software Package). <http://www.mosek.com>.
- [2] D. Bertsimas, D. Brown, and C. Caramanis. Theory and applications of robust optimization. Submitted to SIAM Review.
- [3] D. Collieran, C. Portmann, A. Hassibi, C. Crusius, S. Mohan, S. Boyd, T. Lee, and M. Hershenson. Optimization of phase-locked loop circuit via geometric programming. In *Custom Integrated Circuits Conference*, pages 377–380, 2003.
- [4] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. (web page and software). <http://cvxr.com/cvx>, 2010.
- [5] M. Hershenson. Design of pipeline analog-to-digital converters via geometric programming. In *Proceedings of the International Conference on Computer-aided Design*, pages 317–324, 2002.
- [6] M. Hershenson, S. Boyd, and T. Lee. GPCAD: a tool for CMOS op-amp synthesis. In *Proceedings of the International Conference on Computer-aided Design*, pages 296–303, 1998.
- [7] K. Hsiung, S. Kim, and S. Boyd. Tractable approximate robust geometric programming. *Optimization and Engineering*, 9(2):95–188, 2007.
- [8] J. Kim, J. Lee, and L. Vandenberghe. Techniques for improving the accuracy of geometric-programming based analog circuit design optimization. In *International Conference on Computer Aided Design*, pages 863–870, 2004.
- [9] M. Krasnicki, R. Phelps, R. Rutenbar, and R. Carley. MAELSTROM: efficient simulation-based synthesis for custom analog cells. In *Proceedings of the Design Automation Conference*, pages 945–950, 1999.
- [10] X. Li, P. Gopalakrishnan, Y. Xu, and T. Pileggi. Robust analog/RF circuit design with projection-based performance modeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(1), 2007.
- [11] T. McConaghy, P. Palmers, G. Gielen, and M. Steyaert. Simultaneous multi-topology multi-objective sizing across thousands of analog circuit topologies. In *Proceedings of the Design Automation Conference*, pages 944–947, 2007.
- [12] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley. Synthesis of High-performance Analog Circuits in ASTRX/OBLX. *IEEE Transactions on Computer-aided Design*, 88(12):1825–1852, 2000.
- [13] Y. Xu, K.-L. Hsiung, X. Li, I. Nausieda, S. Boyd, and L. Pileggi. OPERA: Optimization with Ellipsoidal Uncertainty for Robust Analog IC Design. In *Proceedings of the Design Automation Conference*, pages 632–637, 2005.
- [14] S. Boyd, S.-J. Kim, L. Vandenberghe and A. Hassibi. A Tutorial on Geometric Programming. In *Optimization and Engineering*, 8(1):67–127, 2007.