Original scientific paper

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 45, No. 2 (2015), 160 – 170

Computing Worst-Case Performance and Yield of Analog Integrated Circuits by Means of Mesh Adaptive Direct Search

Árpád Bűrmen[1], Husni Habal[2]

[1]University of Ljubljana, Faculty of Electrical Engineering
[2]Technical University of Munich

Abstract: Estimating the parametric yield of a circuit by means of a Monte Carlo analysis can be slow, particularly when the yield estimate is close to 100%, as a large number of samples are necessary to reach the desired level of confidence. Deterministic numerical algorithms have been successfully used in commercial tools for yield estimation. Many of them are gradient-based. The gradients are estimated numerically using finite differences, because most simulators do not compute sensitivities. In this paper, an approach is proposed based on a derivative-free optimization algorithm from the family of mesh adaptive direct search methods. The basic algorithm is extended with capabilities that speed up the convergence and enable the algorithm to cope with infeasible starting points. The new approach is compared to a commercial tool that uses gradient-based algorithms for worst-case analysis. The results show that the proposed approach is capable of producing accurate results within similar computational budgets.

Keywords: analog circuit design;, design centering; worst-case analysis; yield analysis; optimization; mesh adaptive direct search

Določanje najslabših lastnosti in izplena analognih integriranih vezij z adaptivnim mrežnim direktnim optimizacijskim postopkom

Izvleček: Določanje izplena vezja s pomočjo Monte Carlo analize je pogosto zamuden postopek, še posebej, ko se izplen približuje 100%, ker potrebujemo za zanesljive rezultate veliko število vzorcev. Deterministični postopki za določanje izplena so na voljo v komercialnih orodjih. Številni postopki se zanašajo na informacijo o gradientu, ki ga določajo numerično, saj večina simulatorjev ne računa občutljivosti rezultatov. Članek opisuje pristop z uporabo brezgradientnega optimizacijskega postopka iz družine adaptivnih mrežnih direktnih optimizacijskih postopkov. Osnovni postopek je nadgrajen z razširitvami, ki pospešijo konvergenco proti rešitvi problema in omogočajo, da postopek uporabi začetno točko, ki krši omejitve. Predlagani pristop smo primerjali s komercialnih orodjem, ki uporablja gradientne optimizacijske postopke. Rezultati kažejo, da je predlagan pristop sposoben najti pravilne rešitve problemov v primerljivem času.

Ključne besede: načrtovanje analognih vezij; centriranje; določaje najslabših vrednosti lastnosti; analiza izplena; optimizacija; adaptivni mrežni direktni optimizacijski postopki

** Corresponding Author's e-mail: arpadb@fides.fe.uni-lj.si*

1 Introduction

Modern integrated circuits must exhibit adequate performance across a given range of operating conditions, such as supply voltage and temperature, and in the presence of random variations resulting from the manufacturing process [1]. Towards this objective, parametric yield is defined as the fraction of manufactured circuits that meet all performance specifications, such as minimum gain and maximum power, in consideration of all operating conditions, as well as the statistical distribution of random variations. A prerequisite for designing such a circuit is an efficient means of evaluating the circuit's worst performance.

Manufactured circuits that fail an imposed performance specification must be discarded, such that the parametric yield is reduced below 100%. The simplest means to estimate the parametric yield is Monte-Carlo analysis (MCA). Unfortunately, a very large number of performance evaluations are needed for accurate estimation by MCA when the yield is close to 100% [2]. This is prohibitively inefficient, since each performance evaluation requires a costly circuit simulation. More efficient means to evaluate electrical performance given the worst-case combination of operating conditions and random variations is therefore necessary for robust circuit design; some alternatives have been presented in literature (cf. [2, 3]).

In [2], the worst-case distance (WCD) metric was used to obtain yield estimates with less computation. The WCD method requires the numerical solution of an optimization problem. This problem can be solved in significantly less time than it takes an MCA to obtain similar or more accurate yield estimates. The alternative to yield estimation by WCD is the worst-case performance (WCP) method [2]. In WCP, the worst value of a performance is calculated which corresponds to a predefined parametric yield (Y). If this worst value satisfies the performance specification the parametric yield is not smaller than Y.

In general, both WCD and WCP require the solution of a non-linear optimization problem by numerical methods. Deterministic optimizations have been successfully applied to solve the WCD and WCP problems typical in analog integrated circuits -- including academic and commercial tools. These algorithms have been derivative-based, so that the sensitivity of the electrical performances to the value of the operating and statistical parameters was needed (e.g. [2] and the references therein, [3]). In this paper a new deterministic and derivative-free method is proposed to solve the WCD and WCP problems. The method is based on mesh adaptive direct search (MADS)[4].

The remainder of the paper is organized as follows: section 2 introduces the mathematical formulation of WCD and WCP. Section 3 gives an overview of MADS and modifications introduced by the proposed approach. The implementation details are the subject of Section 4. Section 5 presents the results and compares them to the results obtained with a derivative-based algorithm implemented in a commercial tool (WiCkeD [5]). The concluding remarks are given in Section 6.

Notation. Inequalities apply to vectors component wise. ***0*** denotes a vector of all-zeros. The *i-th* component of vector ν is denoted by ν_i. An element of a matrix ***A*** is denoted by a_{ij}. The ramp function *ramp(x)* is zero *for x<0* and equal to *x* otherwise. The group of orthogonal transformations of R^n is denoted by O_n. The *i-th* orthonormal basis vector is denoted by $\boldsymbol{e}_i$.

2 Mathematical formulation for worst-case analysis

Let $\boldsymbol{x}_0$ denote the vector of n_0 parameters describing the circuit's operating condition, also referred to as the operating parameters. The prescribed range of operating conditions within which the circuit must operate is specified by lower and upper bounds on operating parameters given by vectors $\boldsymbol{x}^L_0$ and $\boldsymbol{x}^H_0$, respectively.

The performance of the circuit is also affected by variations of the manufacturing process which in turn are modeled as mutually dependent random variables. Without loss of generality, the set of dependent process parameters can be mathematically transformed into a set of independent random variables with normal distribution. Let $\boldsymbol{x}_S$ denote a vector representing a realization of these random variables. Components of $\boldsymbol{x}_S$ are also referred to as the statistical parameters. By assumption the joint probability density of the statistical parameters can be expressed as

$$p(\boldsymbol{x}_S) = \frac{1}{(2\pi)^{\frac{n_S}{2}}} e^{-\frac{\|\boldsymbol{x}_S\|^2}{2}} \tag{1}$$

Circuit behavior is evaluated by a number of performances, for example power, amplification gain, and bandwidth. The performances are ordered in a vector ***f*** with length m. Their value for any specific circuit will depend on the value of the operating parameters $\boldsymbol{x}_0$, as well as the statistical parameters $\boldsymbol{x}_S$. Component f_i of ***f*** is the value of a map computed by a simulator. With some abuse of notation one can write $f_i(\boldsymbol{x}_0, \boldsymbol{x}_S)$. For a circuit to behave correctly at $(\boldsymbol{x}_0, \boldsymbol{x}_S)$ it must meet a set of performance specifications of the form $f_i(\boldsymbol{x}_0, \boldsymbol{x}_S) \geq G_i$, where G_i denotes the target value corresponding to f_i. Performance specifications of the form $f_i(\boldsymbol{x}_0, \boldsymbol{x}_S) \leq G_i$ can be taken into account by replacing $f_i(\boldsymbol{x}_0, \boldsymbol{x}_S)$ and G_i with $-f_i(\boldsymbol{x}_0, \boldsymbol{x}_S)$ and $-G_i$. A well designed circuit behaves correctly across the given range of operating conditions and for a large percentage of circuits manufactured in the presence of manufacturing process variations (yield). The yield corresponding to performance f_i can be computed by integrating (1) over the acceptance region of f_i (shaded region in Figure 1) defined as

$$f_i^w(\boldsymbol{x}_S) = f_i\left(\boldsymbol{x}_0^{w,i}(\boldsymbol{x}_S), \boldsymbol{x}_S\right) \geq G_i \tag{2}$$

where $f_i^w(\boldsymbol{x}_S)$ is the worst value of f_i at $\boldsymbol{x}_S$ *a*cross the prescribed range of operating parameters and

$$\boldsymbol{x}_0^{w,i}(\boldsymbol{x}_S) = \arg \min_{\boldsymbol{x}_0^L \leq \boldsymbol{x}_0 \leq \boldsymbol{x}_0^H} f_i(\boldsymbol{x}_0, \boldsymbol{x}_S) \tag{3}$$

This integral cannot be computed analytically and is usually estimated with a Monte Carlo analysis.

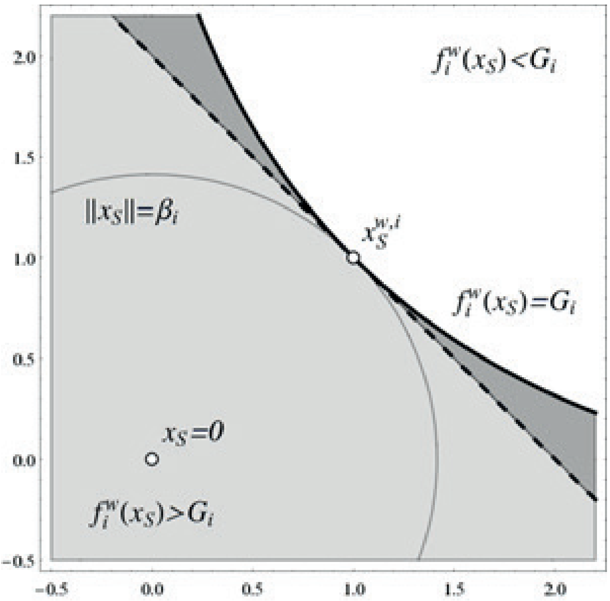

Figure 1: The worst-case point $\boldsymbol{x}_S^{w,i}$ and the linearization (dashed line) of f_i^w $(\boldsymbol{x}_S) = G_i$ (thick line) in the space of the statistical parameters. The acceptance region of f_i is shaded. Replacing the nonlinear specification with its linearization at $\boldsymbol{x}_S^{w,i}$ makes it possible to compute a yield estimate using (6). The approximation introduces an error equal to the integral of (1) over the region shaded in dark grey.

A good yield approximation can be obtained by replacing the performance with its linear model computed at the worst-case point ($\boldsymbol{x}_o^{w,i}$ ($\boldsymbol{x}_S^{w,i}$), $\boldsymbol{x}_S^{w,i}$) [2]. Figure 1 illustrates the worst-case point in the space of statistical parameters when $n_s=2$; the sphere $||\boldsymbol{x}_S||=\beta_i$ is tangential to the boundary of the acceptance region at $\boldsymbol{x}_S^{w,i}$. The statistical parameters corresponding to this point are given by

$$\boldsymbol{x}_S^{w,i} = \begin{cases} \arg \min\limits_{f_i^w(x_S) \le G_i} \|\boldsymbol{x}_S\|^2, for\ f_i^w(\mathbf{0}) \ge G_i \\ \arg \min\limits_{f_i^w(x_S) \le G_i} \|\boldsymbol{x}_S\|^2, otherwise \end{cases} \tag{4}$$

The worst-case distance of f_i is defined as

$$\beta_i = \begin{cases} \|\boldsymbol{x}_S^{w,i}\|, for\ f_i^w(\mathbf{0}) \ge G_i \\ -\|\boldsymbol{x}_S^{w,i}\|, otherwise \end{cases} \tag{5}$$

If $f_i^w(\boldsymbol{x}_S)$ satisfies the design requirements at $\boldsymbol{x}_S = \mathbf{0}$ the worst case distance is positive, otherwise it is negative. By linearizing $f_i^w(\boldsymbol{x}_S) \ge G_i$ at the worst-case point a yield approximation can be computed analytically by integrating (1) over the light grey region in Figure 1. The obtained yield approximation is

$$Y_i = \frac{1}{2}\left(1 + erf\left(\beta_i / \sqrt{2}\right)\right) \approx Y\left(f_i \ge G_i\right) \tag{6}$$

The difference between the actual and estimated yield corresponds to the integral of (1) over the dark grey region in Figure 1.

The computationally intensive component of yield estimation is to find the solution to problem (4). For small yields the computational effort is in the same order of magnitude as that required by a Monte Carlo analysis. For large yields the number of the required Monte Carlo samples grows rapidly as the yield approaches 100% while the computational effort for solving (4) remains the same. Typically a designer tunes the design parameters until β_i (and the yield) is maximized.

Problem (4) has a general nonlinear constraint that can only be evaluated by circuit simulation. An alternative approach to yield maximization is often used. Instead of computing the WCD, the WCP corresponding to a given β can be computed.

$$\left(\boldsymbol{x}_0^{w,i}, \boldsymbol{x}_S^{w,i}\right) = \arg \min_{\substack{\boldsymbol{x}_0^L \le \boldsymbol{x}_0 \le \boldsymbol{x}_0^H \\ \|x_S\|^2 \le \beta^2}} f_i\left(\boldsymbol{x}_0, \boldsymbol{x}_S\right) \tag{7}$$

The constraint in the WCP formulation is a convex quadratic function that can be evaluated without circuit simulation. If the i-th performance f_i satisfies $f_i(\boldsymbol{x}_0^{w,i}, \boldsymbol{x}_S^{w,i}) \ge G_i$, then the WCD ($\beta_i$) and corresponding yield estimate (Y_i) will satisfy $\beta_i \ge \beta$ and $Y_i \ge ½ (1 + \text{erf}(\beta/\sqrt{2}))$.

Problems (4) and (7) are typical problems for which the initial deterministic method of choice is a gradient-based optimization algorithm, for example a sequential quadratic programming (SQP) or an interior point method [6]. An alternative is to use gradient free optimization methods. Mesh adaptive direct search (MADS) is one of these methods. MADS is capable of handling problems with nonsmooth objective function and constraints. Unfortunately as most derivative-free methods MADS converges slowly to a solution. To accelerate its convergence one can use quadratic models of the objective and of the constraints to compute promising points that speed up the algorithm's progress. The quadratic model can be built gradually by applying an update formula to the current approximation of the Hessian matrix.

3 Mesh Adaptive Direct Search

MADS is a family of algorithms where the steps the algorithm takes to explore the search space lie on a grid. In the presented algorithm the grid is defined as

$$\mathcal{G}_k = \left\{ \sum_{i=1}^{n} \Delta_k^m \alpha_i \boldsymbol{e}_i : \alpha_i \in \mathrm{Z} \right\} \tag{8}$$

where D_k^m denotes the mesh size parameter. The algorithm solves problems of the form

$$\min_{\boldsymbol{x} \in \Omega \subseteq \Re^n} f(\boldsymbol{x}) \tag{9}$$

where $\Omega = \{\boldsymbol{x} : \boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^H \wedge c_i(\boldsymbol{x}) \leq 0, \mathrm{i} = 1, \ldots, n_c\}$ denotes the set of feasible points. The lower and the upper bounds on the components of $\boldsymbol{x}$ are given by vectors $\boldsymbol{x}^L$ and $\boldsymbol{x}^H$, respectively. Nonlinear inequality constraints are defined by functions $c_i(\boldsymbol{x})$. For convenience the n_c functions $c_i(\boldsymbol{x})$ are joined in a vector-valued function $\boldsymbol{c}(x)$. The incumbent solution in the k-th iteration and the corresponding value of f are denoted by $\boldsymbol{x}_k$ and f_k, respectively. Any point that is considered to be sufficiently good to replace the incumbent solution is referred to as an improving point. The initial point $\boldsymbol{x}_0 \in \Omega$ corresponds to the first iteration ($k = 0$). MADS can handle constraints with the extreme barrier approach by replacing with $+\infty$, whenever $\boldsymbol{x} \notin \Omega$. Unfortunately this also requires the initial point to be feasible. Infeasible initial points can be handled by using a filter [7][8]. The filter approach decides whether a point can replace the incumbent solution by applying a bi-objective criterion based on the values of the objective and constraints at points evaluated in the past.

Algorithm 1: *k*-th iteration of the proposed algorithm based on the MADS framework.

```
1.  Complete the quadratic model by computing the gradient of $f$ and the Jacobian of $\boldsymbol{c}$.
2.  Make the model convex by replacing the Hessian $\boldsymbol{H}$ with $\boldsymbol{H} + \in \boldsymbol{I}, \in \geq 0$.
3.  Compute $\boldsymbol{s}$ by solving the convex quadratic model and rounding the result to $V_k$.
4.  Evaluate $f$ and $\boldsymbol{c}$ at $\boldsymbol{x} = \boldsymbol{x}_k + \boldsymbol{s}$. If $\boldsymbol{x}$ is an improving point, set $\boldsymbol{x}_{k+1} := x$ and go to step 8.
5.  Generate the set of poll directions $\mathcal{D}_k \subseteq \mathcal{G}_k$.
6.  Evaluate $f$ and $\boldsymbol{c}$ at $\boldsymbol{x} = \theta(\boldsymbol{x}_k + d)$ for $\boldsymbol{d} \in \mathcal{D}_k$.
    If $\boldsymbol{x}$ is an improving point set $\boldsymbol{x}_{k+1} := \boldsymbol{x}$.
    If the step resulting in an improving point was cut, go to step 8, else go to step 7.
    When $D_k$ is exhausted go to step 8.
7.  Evaluate $f$ and $\boldsymbol{c}$ at $\boldsymbol{x} = -\theta(\boldsymbol{x}_k + 2(\boldsymbol{x} - \boldsymbol{x}_k))$. If $\boldsymbol{x}$ is an improving point, set $\boldsymbol{x}_{k+1} := \boldsymbol{x}$.
8.  If $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, set $l_{k+1} := l_k + 1$;
    else if step 7 failed to produce an improving point, set $l_{k+1} := l_k$;
    else if $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$, the step resulting in $\boldsymbol{x}_{k+1}$ was not cut, and $l_k > 0$ set $l_{k+1} := l_k - 1$;
    else set $l_{k+1} := l_k$.
```

Steps 1-4, 5-6, and 7 of Algorithm 1 are also referred to in the MADS literature as the search, the poll, and the speculative step, respectively. Set $\mathcal{D}_k$ is referred to as the set of scaled poll directions. The length of a scaled step is determined by the step size parameter Δ_k^p. Function θ maps points that violate bounds ($\boldsymbol{x}^L$ and $\boldsymbol{x}^H$) to points that satisfy them. A step is cut if $\theta(\boldsymbol{x}) \neq \boldsymbol{x}$. Although the proposed approach uses quadratic models the convergence properties of the MADS framework enable it to find a solution of the optimization problem even when the search step is omitted.

Refining subsequences are sequences of iteration indices $k \in \mathcal{K}$ for which $\Delta_k^p \to 0$. The MADS convergence theory applies to refining subsequences. More details can be found in [4] (extreme barrier approach) and [7] (filter-based approach).

Algorithm 1 differs from the basic MADS framework published in the literature ([4][7][9]) in several ways. The normalized poll directions are uniformly distributed on the unit sphere. The algorithm constructs a quadratic model of the objective function using a minimum Frobenius norm-based approach and a linear model of the constraints by means of regression. A quadratic programming solver then uses the model to compute a search step that accelerates the convergence. The point acceptance criterion in the search and the poll step is based on a filter instead on strict descent.

The algorithm that generates the ordered poll steps and the definition of function θ are the subject of section 3.1. The construction of the quadratic model and a more detailed description of the search step are given in section 3.2. The conditions under which a point is considered to be an improving point are given in section 3.3. The relation between the mesh index (l_k), the mesh size, and the step size is the subject of section 3.4.

3.1 The poll step and the set of scaled poll directions

The poll step (steps 4-6 of Algorithm 1) is the one that guarantees the convergence properties of MADS [4][7]. The scaled poll directions $\boldsymbol{d} \in \mathcal{D}_k$ are ordered according to the angle they form with the last search ($\boldsymbol{s}$) or poll ($\boldsymbol{d}$) direction that resulted in an improving point [9]. The function and the constraints are evaluated at points $\boldsymbol{x}_k + \boldsymbol{d}$ corresponding to the ordered scaled poll directions. If $\boldsymbol{x}_k + \boldsymbol{d}$ violates any of the bounds imposed by $\boldsymbol{x}^L$ and $\boldsymbol{x}^H$ it is replaced by $\theta(\boldsymbol{x}_k + \boldsymbol{d})$. Function θ modifies the components of $\boldsymbol{x}_k + \boldsymbol{d}$ that violate bounds by replacing them with the value of the corresponding violated bound. This has the effect of sliding the point along the violated boundary. The evaluation of points in the poll step is interrupted as soon as an improving point is found (greedy evaluation).

The set of scaled poll directions $\mathcal{D}_k$ is generated by applying an orthogonal transformation $\boldsymbol{O}_{tk} \in \mathcal{O}_n$ to $n + 1$ vectors forming a regular n simplex ν (cf. [10] on how to construct ν) and scaling the resulting vectors with Δ_k^p. This results in set $\mathcal{U}_k = \{\Delta_k^p \boldsymbol{O}_{tk} \boldsymbol{v}{:}\boldsymbol{v} \in \mathcal{V}\}$ whose members are rounded to the nearest points in $\mathcal{G}_k$ to obtain $\mathcal{D}_k$. Index t_k plays a role in ensuring the convergence properties of the algorithm and will be discussed later.

The sequence of orthogonal transformations $\{\boldsymbol{O}_i\}_{i=0}^{\infty}$ is constructed by Algorithm 2 from a sequence of realizations of a random matrix with independent normally distributed random elements $\{\boldsymbol{N}_i\}_{i=0}^{\infty}$.

Algorithm 2: Constructing a sequence of orthogonal transformations [11].

1. Apply QR decomposition to $\boldsymbol{N}_i$ resulting in $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$.
2. Construct diagonal matrix $\boldsymbol{D}_i$ with $d_{ii} = 1$ if $r_{ii} \geq 0$ and $d_{ii} = -1$ otherwise.
3. $\boldsymbol{Q}_i = \boldsymbol{Q}_i \boldsymbol{D}_i$.

Sequence $\{\boldsymbol{O}_i\}_{i=0}^{\infty}$ is uniformly distributed (i.e. distributed according to the Haar measure on $\mathcal{O}_n$ [11]). Due to this the normalized vectors from the sequence of sets $\{\mathcal{U}_i\}_{i=0}^{\infty}$ are uniformly distributed (and consequently dense) on the unit sphere. Furthermore, if the mesh size parameter satisfies $\Delta_k^m \rightarrow 0$ the union of sets $\mathcal{D}_k$ is also dense on the unit sphere (which is required by the MADS convergence theorem [4]) and the distribution of normalized poll directions converges to the uniform distribution on the unit sphere.

3.2 The quadratic model and the search step

MADS can be significantly improved if steps 1-4 of Algorithm 1 examine points obtained by solving a model of the original optimization problem. In the presented method a quadratic model of the objective and a linear model of the constraints are constructed. The model can be formulated as

$$m_f(\boldsymbol{x}) = \frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{x}_k\right)^T \boldsymbol{B}\left(\boldsymbol{x}-\boldsymbol{x}_k\right) + \boldsymbol{g}^T\left(\boldsymbol{x}-\boldsymbol{x}_k\right) + f\left(\boldsymbol{x}_k\right) \tag{10}$$

$$m_c(\boldsymbol{x}) = \boldsymbol{J}\left(\boldsymbol{x}-\boldsymbol{x}_k\right) + c\left(\boldsymbol{x}_k\right) \tag{11}$$

Where $\boldsymbol{B}$, $\boldsymbol{g}$, and $\boldsymbol{J}$ denote the approximate Hessian and the approximate gradient of the objective $f(\boldsymbol{x})$, and the approximate Jacobian of the constraints $\boldsymbol{c}(\boldsymbol{x})$, respectively. The model optimization problem can now be written as

$$\arg\min_{\substack{m_c(\boldsymbol{x}) \leq 0 \\ \boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^H}} m_f(\boldsymbol{x}) \tag{12}$$

The approximate Hessian matrix is obtained by repeatedly applying an update formula. The initial Hessian approximation is set to an all-zero matrix. Every time the algorithm evaluates three collinear points $\boldsymbol{x}$, $\boldsymbol{x} + \alpha_+\boldsymbol{p}$, and $\boldsymbol{x} + \alpha\boldsymbol{p}$ (i.e. after every speculative step that does not violate the bounds) the directional second derivative can be approximated as

$$\frac{d^2 f(\boldsymbol{x}+t\boldsymbol{p})}{dt^2} \approx f_p = \frac{2}{\alpha_+ - \alpha_-}\left(\frac{f(\boldsymbol{x}+\alpha_+\boldsymbol{p}) - f(\boldsymbol{x})}{\alpha_+} - \right. \\ \left. - \frac{f(\boldsymbol{x}+\alpha_-\boldsymbol{p}) - f(\boldsymbol{x})}{\alpha_-}\right) \tag{13}$$

Let $\boldsymbol{B}$ and $\boldsymbol{B}_+$ denote the approximate Hessian and its updated value. When the second directional derivative is available the Hessian update formula from [12] can be used.

$$\boldsymbol{B}_+ = \boldsymbol{B} + \left(f_{\boldsymbol{p}} - \boldsymbol{p}^T\boldsymbol{B}\boldsymbol{p}\right)\frac{\boldsymbol{p}\boldsymbol{p}^T}{\left\|\boldsymbol{p}\right\|^4} \tag{14}$$

It is more common the points are not collinear. In that case an update technique based on least Frobenius norm updating (LFNU) is used [13]. The proposed algorithm uses is a special case of LFNU for $n + 2$ points. It is applied every time a new point is evaluated to update the Hessian of the objective f.

The linear part of the model is computed by means of linear regression [14]. Up to 2n+1 most recently evaluated points ($\boldsymbol{x}$) satisfying $\|\boldsymbol{x} - \boldsymbol{x}_k\| \leq \rho\Delta_k^p$ are selected for regression. The regression computes vector $\boldsymbol{g}$ for which $m_f(\boldsymbol{x})$ is the closest fit to $f(\boldsymbol{x})$ at the selected points. Similarly the approximate Jacobian $\boldsymbol{J}$ of the constraints is obtained by fitting $\boldsymbol{m}_c(\boldsymbol{x})$ to $\boldsymbol{c}(\boldsymbol{x})$.

Whenever a quadratic model of the problem is successfully computed (i.e. the Hessian update and the linear regression are successful) it is used for ordering the scaled poll directions instead of the smallest angle criterion. The primary criterion for model-based direction ordering is the cumulative constraint violation computed as the sum of squares of positive components in vector $\boldsymbol{m}_c(\boldsymbol{x})$. The secondary criterion is the value of the quadratic model $\boldsymbol{m}_f(\boldsymbol{x})$.

The obtained model is used for computing a trial point for the quadratic search step (step 1 of Algorithm 1). For this purpose problem (12) is solved using a quadratic program solver [15]. The solver can handle only positive definite Hessians matrices. Therefore $\boldsymbol{B}$ is replaced with $\boldsymbol{B} + \in\boldsymbol{I}$ and an additional constraint of the form $\|\boldsymbol{x} - \boldsymbol{x}_k\|_\infty \leq \rho$_$\Delta_k^p$ is imposed whenever $\boldsymbol{B}$ is not positive definite. The value of $\in\, > 0$ is chosen by repeatedly applying Cholesky decomposition to $\boldsymbol{B} + \in \boldsymbol{I}$ for increasing values of until the decomposition succeeds [6].

3.3 Point acceptance criterion

When the initial point $\boldsymbol{x}_0$ is feasible a point $\boldsymbol{x}$ can be considered as improving if it is feasible and $f(\boldsymbol{x}) < f(\boldsymbol{x}_k)$. Often the initial point $\boldsymbol{x}_0$ is not feasible (i.e. $\boldsymbol{x}_0 \notin \Omega$). Such a situation occurs when one tries to solve (4) to obtain the worst-case distance and chooses **0** as the initial value of the statistical parameters. In this case the nonlinear constraints cannot be handled with the extreme barrier approach. A possible alternative is to use a point acceptance criterion based on a filter [8].

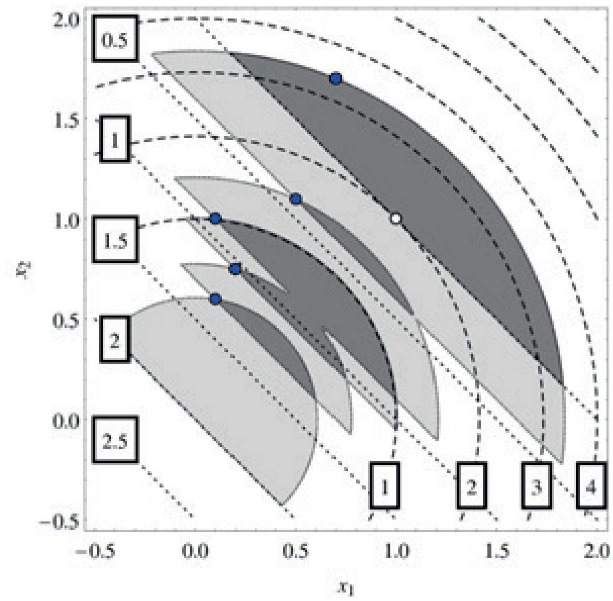

Figure 2: The regions with acceptable (light gray) and dominating (dark gray) points for an optimization problem given by $f(x_1, x_2) = x^2_1 + x^2_2$ (dashed contours) and $c(x_1, x_2) = -x_1 - x_2 + 2$ (dotted contours). The points corresponding to the filter entries are marked by dark dots. The white dot marks the solution of the problem at (1, 1). $h_{max} = 2$.

The acceptance criterion based on a filter takes into account an improvement of the objective value, as well as an improvement of the feasibility. For that purpose a function is defined that expresses the constraint violation.

$$h(\boldsymbol{x}) = \sum_{i=1}^{n_c} ramp\left(c_1(\boldsymbol{x})\right) \qquad (15)$$

For a feasible point the corresponding value of $h(\boldsymbol{x})$ is zero. A filter entry is a tuple of the form $(f(\boldsymbol{x}), h(\boldsymbol{x}))$. A filter is a set of mutually non-dominated filter entries. A tuple (f_1, h_1) dominates (f_2, h_2) if $f_1 \leq f_2$, $h_1 \leq h_2$ and the two tuples are not equal. Initially the filter contains only $(f(\boldsymbol{x}_0), h(\boldsymbol{x}_0))$. A point $\boldsymbol{x}$ is said to be

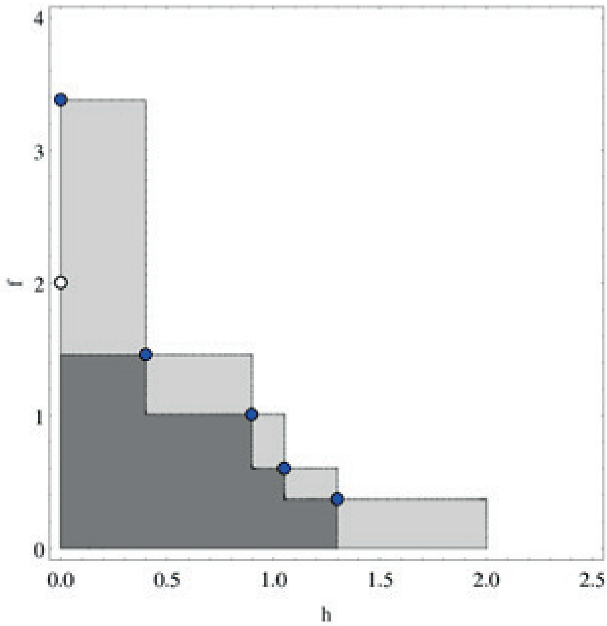

Figure 3: The filter entries (dark dots) and the solution (white dot) of the problem in Figure 2 in the *f-h* space. Dark gray and light gray regions correspond to dominating and acceptable points, respectively.

- dominating if the filter is empty or $(f(\boldsymbol{x}), h(\boldsymbol{x}))$ dominates at least one filter entry,
- dominated if at least one filter entry dominates $(f(\boldsymbol{x}), h(\boldsymbol{x}))$ or $h(\boldsymbol{x}) > h_{max}$.
- acceptable otherwise.

Figure 2 and Figure 3 illustrate a 2-dimensional problem and a filter with 5 entries. Adding a point to the filter means that the corresponding filter entry $(f(\boldsymbol{x}), h(\boldsymbol{x}))$ is added to the filter. Dominating points and acceptable points are always added to the filter immediately after they are evaluated. The incumbent solution is always a member of the filter. Adding a dominating point implies that at least one dominated point is removed from the filter so that the filter entries remain mutually non-dominated. An acceptable point does not dominate any of the filter points and thus no points are removed from the filter when the corresponding entry is added. Dominated points are never added to the filter. If parameter h_{max} is set to 0, MADS behaves as if the extreme barrier approach had been used for handling the constraints.

For every filter point its position is defined by sorting the filter entries according to *h*. The filter entry corresponding to a feasible point is assigned position 0 (rightmost dark dot in Figure 2, leftmost dark dot in Figure 3), while infeasible filter entries are assigned integer positions starting from 1. A point examined by the search step is considered as an improving point if it is not dominated. A point examined in the poll step and in the speculative step is considered as an improving point if it is not dominated and its position is not higher than the position of the incumbent solution. This effectively requires that the poll and the search step prioritize improving feasibility over improving the objective. When the incumbent solution is feasible these two steps behave as if the extreme barrier approach had been used.

3.4 Updating the mesh and the step size

Iterations of Algorithm 1 are assigned a mesh index l_k with initial value $l_0 = 0$. The mesh and the step size parameter depend on l_k.

$$\Delta_k^m = \min\left(1, \Delta^{-2l_k}\right) / \left(\Delta_0 \lceil 1 + \gamma \rceil\right) \tag{16}$$

$$\Delta_k^p = \Delta^{-l_k} \tag{17}$$

This strategy (see step 8 of Algorithm 1) refines the mesh and shortens the step when the algorithm is not making progress (i.e. fails to find an improving point). The mesh index is not changed if the speculative step fails to produce an improving point or if the improving point is obtained with a cut step. Otherwise the mesh is coarsened and the step size is increased, but never above its initial value.

Rounding can affect the set of unrounded scaled poll directions U_k to such extent that $\mathcal{D}_k$ no longer positively spans $\mathbb{R}^n$. The effect of rounding is more pronounced when ratio Δ^p_k/Δ^m_k is small. Because

$$\frac{\Delta_k^p}{\Delta_k^m} = \Delta_0 \lceil 1 + \gamma \rceil 2^{|l_k|} \geq \Delta_0 \lceil 1 + \gamma \rceil \tag{18}$$

the aforementioned situation can be avoided if one chooses a sufficiently large . It can be shown that $\gamma = n^{3/2}/2$ is an appropriate choice for all $\Delta_0 \geq 1$.

The normalized poll directions from a refining subsequence $\{\mathcal{D}_k\}_{k\in\mathcal{K}}$ must be dense on the unit sphere [4]. This is true if the refining subsequence corresponds to the complete sequence $\{\boldsymbol{N}_i\}^{\infty}_{i=0}$.Therefore index is chosen in the following manner.

$$t_k = \begin{cases} l_k, \, for \; l_k \geq \max\limits_{i\langle k} l_i \\ 1 + \max\limits_{i\langle k} t_i, \text{ othervise} \end{cases} \tag{19}$$

Index t_k increases from iteration to iteration with the exception of iterations that correspond to the finest observed mesh over 0..*k*. As the mesh index of iterations forming a refining subsequence takes consecutive values from {0,1,2, …} the same values are also assigned to thereby causing $\{\mathcal{D}_k\}_{k\in\mathcal{K}}$ to correspond to the complete sequence $\{\boldsymbol{N}_i\}^{\infty}_{i=0}$.

4 Finding the worst-case point

The outline of the proposed approach comprises the same steps as [5]. The SQP-based optimization algorithm is replaced with the proposed version of MADS. The initial point in the space of statistical parameters is computed from the linearized optimization problem. An extended stopping criterion is proposed based on the approximate gradient of the circuit's performance.

In the beginning $\boldsymbol{x}_s = \boldsymbol{0}$, $\boldsymbol{x}_0$ is set to the nominal value of the operating parameters $\boldsymbol{x}_0^{nom}$, and the set of relevant statistical parameters is empty. The procedure for solving problem (4) and problem (7) is given by Algorithm 3.

Algorithm 3: One pass of the main algorithm for solving problem (4) / problem (7)

Solve $\boldsymbol{x}_0^{w2} = \arg\min_{\boldsymbol{x}_0^L \leq \boldsymbol{x}_0 \leq \boldsymbol{x}_0^H} f_i\left(\boldsymbol{x}_0, \boldsymbol{x}_S\right)$. If $f_i\left(\boldsymbol{x}_0, \boldsymbol{x}_0^{w2}\right)$, set $\boldsymbol{x}_0 := \boldsymbol{x}_0^{w2}$.

1. Compute the approximate sensitivity of f_i to statistical parameters.
2. Update the set of relevant statistical parameters and compute the initial point for step 4.
3. Solve (4) or (7) in the space of relevant statistical parameters to obtain the new value of $\boldsymbol{x}_S$.

In step 1 of Algorithm 3 the set of worst operating parameters is determined. The performance corresponding to ($\boldsymbol{x}_0$, $\boldsymbol{x}_S$) is evaluated. Every operating parameter is perturbed to its respective upper and lower value resulting in the need to evaluate $2n_0$ points by circuit simulation. The results are used for constructing the initial vector of operating parameters ($\boldsymbol{x}_0^{w1}$). Every component of this vector is equal to the nominal value, the upper bound, or the lower bound of the corresponding operating parameter, whichever produced the worst value of f_i. The optimization in step 1 of Algorithm 3 is completed with the MADS algorithm as proposed in section 3 starting from $\boldsymbol{x}_0^{w1}$ and using the extreme barrier approach. Steps taken by the optimizer are scaled in such manner that a step of length 1 in direction of any operating parameter corresponds to 1/16 of the difference between the upper and the lower bound.

The sensitivity to the statistical parameters (step 2 of Algorithm 3) is computed at ($\boldsymbol{x}_0$, $\boldsymbol{x}_S$) using forward differences. The parameters are perturbed by 1/64 of the difference between the lower and the upper bound (-10 and 10, respectively). Let Δx and Δf_i denote the parameter perturbation and the corresponding difference in the performance, respectively. The components of the gradient with respect to the statistical parameters ($\nabla_S f_i$) can then be approximated as $\Delta f_i / \Delta x$.

The obtained sensitivity information is used for eliminating the statistical parameters that contribute little to the behavior of f_i (step 3 of Algorithm 3). For this purpose the absolute performance differences $|\Delta f_i|$ are ordered and all parameters that contribute less than 1% to the total change of f_i are removed in increasing order of contribution until the cumulative contribution of the removed parameters reaches 25% or there are no statistical parameters left. The remaining statistical parameters are added to the set of relevant statistical parameters.

Let $\boldsymbol{g}_S$ denote the estimated gradient in the space of statistical parameters. Components of the gradient not corresponding to relevant parameters are set to 0. For problem (4) the initial point is obtained by updating $\boldsymbol{x}_S$ to

$$\boldsymbol{x}_S - \frac{f_i(\boldsymbol{x}_0,\mathbf{0}) - G_i}{\|\boldsymbol{g}_S\|^2}\boldsymbol{g}_S \tag{20}$$

For problem (7) x_S is replaced with

$$-\frac{\boldsymbol{g}_S}{\|\boldsymbol{g}_S\|^2}\beta \tag{21}$$

MADS is then used for solving the main optimization problem (step 4 of Algorithm 3) in the space of statistical parameters. The value of h_{max} is chosen as max(100, $h(\boldsymbol{x}_0)$) so that the initial point is always added to the filter. The scaling of parameters is the same as in step 1 of Algorithm 3. The main optimization in case of problem (7) is stopped if the constraint satisfaction condition $|c(\boldsymbol{x})| < \beta\varepsilon_c$ and the gradient angle condition $\angle(\nabla_S f, -\nabla_S c) < \varepsilon_a$ are satisfied (note that $c(\boldsymbol{x})$ is a scalar because the problem has only one nonlinear constraint). These two stopping conditions are applied only if the step satisfies $\Delta^p_k < 0.5$. The constraint satisfaction condition for problem (4) is formulated somewhat differently as $|c(\boldsymbol{x})| < 3||\boldsymbol{g}_S||\varepsilon_c$. In the presented examples $\varepsilon_c = 10^{-2}$ and $\varepsilon_a = 15^\circ$ are used. Regardless of these stopping conditions MADS is stopped when Δ^p_k drops below 0.01.

Algorithm 3 is repeated in multiple passes until the set of relevant statistical parameters remains unchanged in step 3 and the accepted solution in step 1 of Algorithm 3 does not change f_i ($\boldsymbol{x}_0$, $\boldsymbol{x}_S$) by more than 1% compared to the difference between f_i ($\boldsymbol{x}_0^{nom}$, $\mathbf{0}$) and f_i ($\boldsymbol{x}_0$, $\mathbf{0}$) from the first pass.

The following values of optimizer parameters were used: $\Delta = 4$, $\Delta_0 = 2^{20}$, $\rho = \Delta^2$, $\rho\_ = 1$. For problem (7) the gradient of the constraint with respect to the statistical parameters can be expressed explicitly as $2\boldsymbol{x}_S$ and is not computed numerically. Similarly for problem (4) the gradient and the Hessian of the objective can be expressed as $2\boldsymbol{x}_S$ and $\boldsymbol{I}$ (identity matrix), respectively.

5 Application and verification of the approach

The proposed approach was implemented in the PyOPUS framework [16] and its performance was compared to that of a commercial worst-case analysis tool WiCkeD [5]. Both algorithms were tested on two circuits: a Miller operating transconductance amplifier (OTA) in Figure 4 and a folded cascode operating transconductance amplifier (FCOTA) in Figure 5.

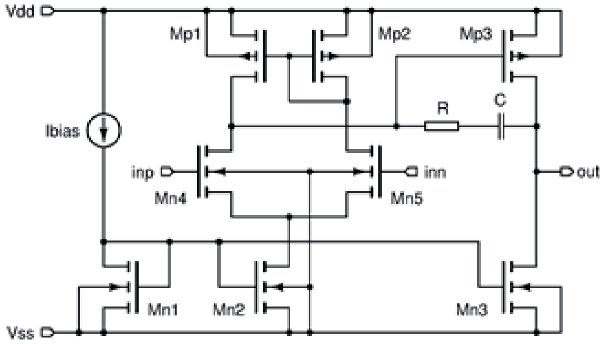

Figure 4: Miller transconductance amplifier.

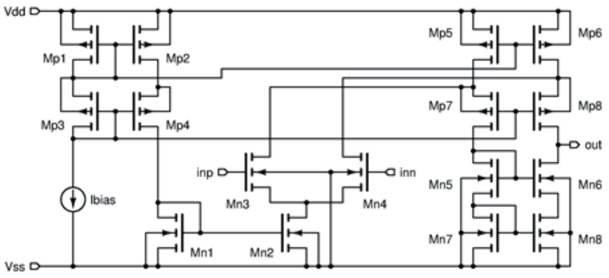

Figure 5: Folded cascode transconductance amplifier.

Both circuits have 3 operating parameters (temperature, supply voltage, and bias current). A mismatch model with two statistical parameters per transistor was used. Global variations of the manufacturing process were modeled with 10 statistical parameters. The circuits in Figure 4 (Figure 5) have 26 (42) statistical parameters. The results are listed in Table 1. The first and the second column list the names of the performances and their types (i.e.$f_i > G_i$ or $f_i < G_i$). The worst-case performances at $\beta = 3$ obtained by solving problem (7) are listed in columns titled WC. The number of circuit evaluations and the number of algorithm passes are listed in the columns to the right of the WC column.

Problem (4) is solved with G_i set to the WC value at $\beta = 3$ obtained by WiCkeD (third column). The worst-case point obtained by solving this problem lies at $||\boldsymbol{x}_s|| = 3$. The obtained value of b_i is listed in columns titled WCD and the number of circuit evaluations and algorithm passes is listed in the columns to the right of the WCD column.

The results in Table 1 show that the proposed approach is capable of finding the solution of problem (7) within 5% accuracy. The two cases where the accuracy was worse than 5% are marked with an asterisk in the WC column. The settling time (rise) of the Miller OTA was different due to the noise in the performance. In case of the PSRR VSS performance of the FCOTA circuit MADS converged to a different local minimizer. A more pessimistic worst case value was found by MADS in one case (shaded cell in the table). The number of circuit evaluations required by MADS was in 7 cases (marked with an asterisk) significantly worse than that required by WiCkeD. On the other hand in two cases MADS was significantly faster than WiCkeD (shaded cells in the table). On the remaining cases both algorithms exhibited similar performance.

Solving problem (4) is somewhat more challenging. The proposed approach found the same solution within 5%

Table 1: Summary of the results obtained with WiCkeD and the proposed MADS-based algorithm. A WC/WCD value (the number of evaluations) that is more than 5% (20%) worse than the corresponding result obtained by WiCkeD is denoted by an asterisk.

		WiCkeD				MADS					
Circuit / Performance	type	WC	Evals	WCD	Evals	WC	Evals	Passes	WCD	Evals	Passes
Miller OTA											
Swing [V]	>	1.43	139	2.99	145	1.43	147	2	3.00	*176	2
Gain [dB]	>	68.0	88	3.00	94	68.0	98	1	3.00	106	1
UGBW [MHz]	>	1.61	93	3.00	100	1.61	98	1	3.02	116	1
Phase margin [°]	>	67.3	129	3.00	123	67.3	*299	2	3.04	*438	2
CMRR [dB]	>	65.2	98	3.00	104	65.3	*150	2	3.00	*166	2
PSRR VDD [dB]	>	85.0	124	3.00	112	85.3	*396	3	*3.21	*398	3
PSRR VSS [dB]	>	61.0	92	3.00	98	61.0	98	1	3.00	106	1
Settling ↓ [μs]	<	0.892	134	3.00	151	0.892	145	2	3.01	165	2
Settling ↑ [μs]	<	1.04	108	3.00	116	*1.03	102	1	3.00	*195	2
Slew ↓ [V/μs]	>	1.10	94	3.00	96	1.10	99	1	3.00	115	1
Slew ↑ [V/μs]	>	0.953	461	3.06	196	0.960	101	1	3.09	*260	2
FCOTA											
Offset (high) [mV]	<	11.2	124	3.02	194	11.2	*202	2	3.00	211	2
Offset (low) [mV]	>	-11.9	124	3.03	194	11.9	*200	2	3.00	231	2
Swing [V]	>	0.478	122	3.00	127	0.476	130	1	2.95	130	1
Gain [dB]	>	70.7	125	3.03	131	70.8	*291	2	3.02	*332	2
UGBW [MHz]	>	6.28	130	3.00	137	6.28	133	1	3.00	149	1
Phase margin [°]	>	85.6	222	3.00	227	85.6	*368	2	3.01	*493	3
CMRR [dB]	>	60.8	290	3.06	265	60.8	*460	2	3.00	*456	2
PSRR VDD [dB]	>	55.8	236	3.00	207	58.0	282	2	3.12	*642	2
PSRR VSS [dB]	>	54.6	315	3.00	227	*58.3	249	2	*3.17	*315	2
IRN@100Hz [μV/ √Hz]	<	3.17	126	3.00	133	3.17	133	1	3.01	151	1
IRN@10kHz [μV/ √Hz]	<	0.321	126	3.00	133	0.321	133	1	3.02	154	1
IRN@1MHz [μV/ √Hz]	<	59.4	126	3.03	132	59.3	133	1	3.00	144	1
1/f corner [kHz]	<	437	122	3.00	128	436	143	1	3.01	147	1
Settling ↓ [μs]	<	0.131	142	3.00	148	0.131	135	1	3.00	146	1
Settling ↑ [μs]	<	0.127	148	3.00	154	0.127	134	1	3.01	180	1
Slew ↓ [μs]	>	4.17	207	3.00	202	4.17	203	2	3.00	*250	2
Slew ↑ [μs]	>	4.28	196	3.00	213	4.28	205	2	3.00	255	2

accuracy in all but two cases marked with an asterisk in the WCD column of Table 1. Both of them (as well as the PSRR VDD performance of FCOTA) are the result of convergence to a different local minimizer. In such cases a fair comparison with WiCkeD is not possible. When the number of circuit evaluations is considered both approaches exhibit similar performance on more than half of the performances. The cases where the proposed approach is significantly slower than WiCkeD are marked with an asterisk.

All optimization problems except for two are solved in one or two algorithm passes. Both MADS and WiCkeD face the same disadvantage originating from the local nature of the underlying optimization algorithms. Due to it the obtained solution can be a local minimizer and not the actual solution of problem (4) or (7) because the outcome greatly depends on the choice of the initial point.

MADS performs best on noisy nonlinear problems for which points exist where the function or the constraints cannot be evaluated (i.e. the simulator fails to converge or the circuit's performance cannot be evaluated). For such problems the finite difference approximation of the gradient cannot be computed and classical optimization methods like SQP used in commercial tools exhibit slow progress or fail. On these problems we expect MADS to outperform commercial gradient-based tools.

6 Conclusion

Finding the worst performance and the worst-case distance of a circuit's performance are important subproblems that arise in the process of automated integrated circuit sizing. The solution to these problems enables the designer to verify the satisfaction of the minimum yield requirement. This is an accurate and less costly alternative to yield estimation by Monte-Carlo analysis. An approach for solving both problems by means of MADS was presented. Several extensions were implemented in the general MADS framework that make it possible for the algorithm to rapidly close in on the solution of the optimization problem. The proposed algorithm was tested on two real world integrated circuit design problems. The results were compared to the results obtained with a commercial worst-case analysis tool (WiCkeD) that uses a gradient-based optimization algorithm. The results show the proposed approach is competitive with the approach used in WiCkeD.

7 Acknowledgements

The research was co-funded by the Ministry of Education, Science, and Sport (Ministrstvo za Šolstvo, Znanost in Šport) of the Republic of Slovenia through the programme P2-0246 Algorithms and optimization methods in telecommunications.

8 References

1. K. Papathanasiou, "A designer's approach to device mismatch: Theory, modeling, simulation techniques, scripting, applications and examples", Analog Integrated Circuits and Signal Processing, vol. 48, no. 2, pp. 95-106, 2006.
2. H. E. Graeb, "Analog design centering and sizing", Springer, 2007.
3. A. Singhee, R. A. Rutenbar (eds.), "Extreme Statistics in Nanoscale Memory Design", Springer, 2010.
4. C. Audet, J. E. Dennis, Jr., "Mesh adaptive direct search algorithms for constrained optimization", SIAM Journal on Optimization, vol. 17, no. 1, pp. 188–217, 2006.
5. MunEDA inc, "WiCkeD, a tool suite for nominal and statistical custom IC design", available at http://www.muneda.com/Products/, 2014.
6. J. Nocedal, S.Wright, "Numerical optimization", Springer, 2006.
7. C. Audet, J. E. Dennis, Jr., "A progressive barrier for derivative-free nonlinear programming", SIAM Journal on Optimization, vol. 20, no. 1, pp. 445–472, 2009.
8. R. Fletcher, S. Leyffer, "Nonlinear programming without a penalty function", Mathematical Programming, vol. 91, no. 2, pp. 239–270, 2002.
9. C. Audet, A. Ianni, S. Le Digabel, C. Tribes, "Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms", SIAM Journal on Optimization, vol. 24, no. 2, pp. 621–642, 2014.
10. A. R. Conn, K. Scheinberg, L. N . Vincente, "Introduction to derivative-free optimization", SIAM, 2009.
11. G. W. Stewart, "The efficient generation of random orthogonal matrices with an application to condition estimators", SIAM Journal on Numerical Analysis, vol. 17, no. 3, pp. 403–409, 1980.
12. D. Leventhal, A. S. Lewis, "Randomized Hessian estimation and directional search", Optimization: A Journal of Mathematical Programming and Operations Research, vol. 60, no. 3, pp. 329–345, 2011.
13. M. J. D. Powell, "Least Frobenius norm updating of quadratic models that satisfy interpolation conditions", Mathematical Programming, vol. 100, no. 1, pp. 183–215, 2003.

14. A. L. Custódio, L. N. Vincente, "Using sampling and simplex derivatives in patters search methods", SIAM Journal on Optimization, vol. 18, no. 2, pp. 537–555, 2007.
15. M. S. Andersen, J. Dahl, L. Vandenberghe, "CVXOPT, Release 1.1.6", available at http://cvxopt.org/userguide/index.html, 2014.
16. "PyOPUS - Circuit Simulation and Optimization", available at http://fides.fe.uni-lj.si/pyopus/, 2014.

Arrived: 12. 02. 2015
Accepted: 09. 05. 2015