Journal of Circuits, Systems, and Computers
Vol. 18, No. 7 (2009) 1185–1204
© World Scientific Publishing Company

# A DIRECT SEARCH METHOD FOR WORST CASE ANALYSIS AND YIELD OPTIMIZATION OF INTEGRATED CIRCUITS

GREGOR CIJAN
*Regional Development Agency of Northern Primorska,*
*Mednarodni Prehod 6, SI-5290 Šempeter pri Gorici, Slovenia*
*gregor.cijan@rra-sp.si*

TADEJ TUMA* and ÁRPÁD BŰRMEN†
*Faculty of Electrical Engineering, University of Ljubljana,*
*Tržaška 25, SI-1000 Ljubljana, Slovenia*
**tadej.tuma@fe.uni-lj.si*
†*arpadb@fides.fe.uni-lj.si*

Revised 5 June 2009

Yield maximization is an important aspect in the design of integrated circuits. A prerequisite for its automation is a reliable and fast worst performance analysis which results in corners that can be used in the process of circuit optimization. We formulate the constrained optimization problem for finding the worst performance of an integrated circuit and develop a direct search method for solving it. The algorithm uses radial steps and rotations for enforcing the inequality constraint. We demonstrate the performance of the proposed algorithm on real world design examples of integrated circuits. The results indicate that the algorithm solves the worst performance problem in an efficient manner. The proposed algorithm was also successfully used in the process of yield maximization, resulting in a 99.65% yield.

*Keywords*: Worst-case analysis; MOS transistor mismatch; yield maximization; integrated circuit.

## 1. Introduction

The design of modern analog and mixed-signal integrated circuits (IC) is heavily based on computers. Because a large part of integrated circuits must interact with the environment (which in turn is analog by nature), analog IC design is always present in the design flow. The performance of analog circuits is expressed by means of performance measures (e.g., gain, bandwidth, etc.) which are real numbers. A circuit is said to perform adequately if these performance measures satisfy the design requirements. Design requirements are expressed as inequalities imposed on the performance measures (e.g., bandwidth must be above 10 MHz).

The performance of an analog circuit is affected by three groups of parameters.[1] The first group are the design parameters. They can be adjusted by the designer in the process of IC design. The second group are the range parameters that describe the environment in which the circuit operates. Two very common range parameters are temperature and supply voltage. The circuit is expected to exhibit adequate performance across the whole range of range parameters (e.g., for temperatures between −20°C and 80°C). Finally the third group comprises the so-called statistical parameters. These parameters vary randomly due to the random nature of the manufacturing process. Random variations are the reason why some manufactured circuits do not perform adequately across the whole range of range parameters. The share of circuits that perform adequately across the whole range of range parameters subject to variations of statistical parameters is also referred to as the yield. Considering only one performance measure results in a partial yield for that performance measure.

The goal of robust IC design is to choose the design parameters in such manner that the design requirements are fulfilled across the whole range of range parameters for as many circuits as possible (yield maximization).[2] A prerequisite for its automation is a fast and reliable method for obtaining either the yield itself or a measure of the circuit's worst performance.[3,4]

Direct yield evaluation is a very time-consuming task. Therefore various approaches were developed for fast yield estimation.[5,6] One of the more recent is an approach based on worst case distance (WCD) proposed by Graeb.[7] WCD depends on the design parameters and can be directly translated into a partial yield estimate. Yield can be estimated from the WCD values of performance measures without additional circuit simulations. Obtaining the WCD is an optimization problem with a simple objective and a complicated nonlinear constraint that can only be evaluated by simulating the circuit.

On the other hand, robust design can also be automated using worst performance analysis. The analysis results in worst parameter values for every performance measure. Many of the proposed approaches to worst performance analysis assumed that the mean values and variances of statistical parameters are at the same time the design parameters.[8,9] Such methods are well suited for board-level design, but not for IC design.

In IC design, worst performance analysis results in the values of range and statistical parameters for which a performance measure violates the respective design requirement as much as possible, or at least comes closest to violating it. Generally every performance measure has a corresponding set of worst case parameter values (which in turn depend on the design parameters). The value of a performance measure at the corresponding set of worst case parameters is also referred to as the worst performance measure (WPM). If all WPMs satisfy the design requirements we can safely say that a robust design has been achieved.

To obtain a WCD of a performance measure in an efficient manner, one must be capable of computing the gradient of the performance measure with respect to the statistical and range parameters. The values of performance measures are usually obtained by computer simulation. Most simulators do not calculate the derivatives of the results with respect to the input (statistical and range) parameters. In such a case, gradients can be obtained using finite differences (often with a significant numerical error) or one can resort to so-called direct search optimization methods that require no derivatives.[10] Because most direct search methods cannot handle general nonlinear constraints and still reach a solution in a reasonable time, they are not well suited for the WCD problem.[11,12]

In this paper we propose a direct search method for solving the WPM problem, which is an alternative to the WCD problem. Because of simple constraints that appear in the WPM problem, direct search methods are good candidates for finding its solution. The resulting range and statistical parameter values define corners for the circuit optimization.[13] In turn they can be used to obtain a robust design in a manner similar as Mukherjee did.[14] The main focus of this paper is not on the yield optimization itself. We concentrate on the subproblem of finding the worst performance measure (e.g., the corners for the robust design).

We first formulate the WPM problem and its relation to the WCD problem. We define the underlying optimization problem in Sec. 2. In Sec. 3 we propose a direct search method that is well suited for solving the WPM problem. The method handles the inequality constraint of the WPM problem by choosing appropriate trial steps. Section 4 illustrates the method by finding the corners corresponding to the design requirements of some real-world IC designs. The computational cost of the proposed method is illustrated by listing the equivalent number of finite difference gradient evaluations. We also use the algorithm as the basis for yield maximization of one of the test circuits. Finally, Sec. 5 gives the conclusion.

**Notation.** Real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively. Vectors are assumed to be column vectors and denoted by a bold typeface ($\mathbf{x}$). The dot product of two vectors is written as $\mathbf{x}^T\mathbf{y}$ and the 2-norm of a vector as $\|\mathbf{x}\|$. Components of a vector are denoted by italic typeface with a subscript acting as the component index ($a_i$).

## 2. Problem Definition

### 2.1. *Relevant notions from IC design*

Let $f_i$ denote one of the $m$ performance measures. Every design requirement can then be formulated as $f_i \geq F_i$ (for requirements of the form $f_i \leq F_i$ we simply replace $f_i$ and $F_i$ with $-f_i$ and $-F_i$). $F_i$ is also referred to as the goal. For a predefined circuit topology the performance measures depend on the circuit's parameters. Let $\mathbf{x}_D$, $\mathbf{x}_R$, and $\mathbf{x}_S$ denote the design, range, and statistical parameters. Similarly let $n_D$, $n_R$, and $n_S$ denote the number of design, range, and statistical parameters, respectively.

The range parameters (as the name implies) come from a range of values. Let $\mathcal{R}$ denote the set of possible range parameter values. If, for instance, our range parameters are the temperature and the supply voltage with values from −20°C to 80°C and from 1.6 V to 2.0 V, the set $\mathcal{R}$ is defined as $[-20, 80] \times [1.6, 2.0]$. $\mathcal{R}$ is often referred to as the tolerance body of the range parameters.

Statistical parameters are usually modeled with a normal joint probability distribution function (JPDF).

$$p(\mathbf{x}_S) = \frac{1}{(\sqrt{2\pi})^{n_S}\sqrt{\det \mathbf{C}}} \exp\left[\frac{-\beta^2(\mathbf{x}_S)}{2}\right], \tag{1}$$

where $\beta^2(\mathbf{x}_S) = (\mathbf{x}_S - \mathbf{x}_S^N)^T\mathbf{C}^{-1}(\mathbf{x}_S - \mathbf{x}_S^N)$, $\mathbf{x}_S^N$ is the vector of mean statistical parameter values (also referred to as the nominal statistical parameters) and $\mathbf{C}$ denotes the covariance matrix. If the JPDF is continuous but not normal we can make it normal by applying an appropriate transformation to the statistical parameters. As a matter of fact we can always transform such JPDFs into a normal distribution with $\mathbf{x}_S^N = \mathbf{0}$ and $\mathbf{C} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Therefore without loss of generality we assume the latter distribution throughout the remainder of this paper.

Note that statistical parameters are (unlike range parameters) unbounded. Inequality $\beta(\mathbf{x}_S) \le \beta_i^W$ is the most natural way of defining a tolerance body for a normal JPDF. It defines a so-called tolerance ellipsoid (in our case where $\mathbf{C} = \mathbf{I}$, it is a hypersphere with radius equal to $\beta_i$). Statistical parameters arise from the variations of the manufacturing process also referred to as mismatch. Due to mismatch, two equally designed devices exhibit different electrical behavior and consequently manufactured circuit performances differ from their desired values in a random manner.[15–18] Mismatch effects can be divided in two components: stochastic and systematic. The stochastic component can be reduced with either better process control (usually not an option for a circuit designer) or trade-offs (with circuit area) in the design stage while the systematic component can be reduced in an effective way with proper layout.[19, 20]

The circuit's performance measures can now be written as $f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S)$. Statistical parameters are the reason why not all manufactured circuits satisfy the design requirements across all $\mathbf{x}_R \in \mathcal{R}$. The share of manufactured circuits that satisfy one particular design requirement is called yield partition and can be obtained as

$$Y_i(\mathbf{x}_D) = \int_{\mathbf{x}_S \in \mathbb{R}^{n_S}} p(\mathbf{x}_S) h_i(\mathbf{x}_D, \mathbf{x}_S) \mathrm{d}\mathbf{x}_S\,, \tag{2}$$

where $h_i(\mathbf{x}_D, \mathbf{x}_S)$ is the indicator function of $i$th performance measure (i.e., it is 1 if $f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S) \ge F_i$ for all $\mathbf{x}_R \in \mathcal{R}$ and 0 otherwise). The set $\{\mathbf{x}_S : h_i(\mathbf{x}_D, \mathbf{x}_S) = 1\}$ is also referred to as the acceptance region of $i$th performance measure. The acceptance region and the yield partition are both functions of the design parameters.

Total yield is the share of manufactured circuits that satisfy all design requirements across all $\mathbf{x}_R \in \mathcal{R}$. It is defined in a way similar to Eq. (2) except that $h_i(\mathbf{x}_D, \mathbf{x}_S)$ is replaced by $\prod_{i=1}^m h_i(\mathbf{x}_D, \mathbf{x}_S)$. As all yield partitions approach 1 the total yield also approaches 1.

### 2.2. *The WCD and the WPM problem*

Graeb has shown that a yield partition ($Y_i$) can be closely approximated using WCD ($\beta_i(\mathbf{x}_D)$), which in turn is defined as

$$\beta_i(\mathbf{x}_D) = \min \beta(\mathbf{x}_S) \quad \text{subject to} \min_{\mathbf{x}_R \in \mathcal{R}} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S) \le F_i \,, \tag{3}$$

if $\min_{\mathbf{x}_R \in \mathcal{R}} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S^N) \ge F_i$ and

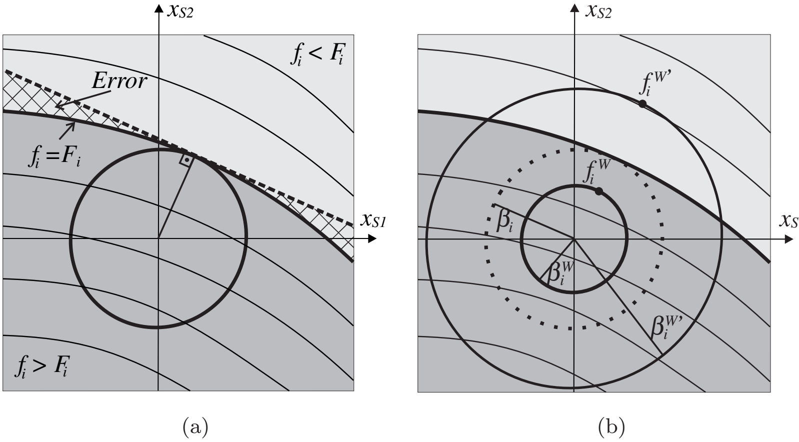
$$\beta_i(\mathbf{x}_D) = -\min \beta(\mathbf{x}_S) \quad \text{subject to} \min_{\mathbf{x}_R \in \mathcal{R}} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S) \ge F_i \,, \tag{4}$$

otherwise.[1] Figure 1(a) illustrates the WCD problem for a positive WCD. The WCD ($\beta_i$) is obtained by finding the point where contour $\min_{\mathbf{x}_R \in \mathcal{R}} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S) = F_i$ touches the sphere $\beta(\mathbf{x}_S) = \beta_i$ with the smallest radius. The distance of the point from the origin is the WCD.

A good approximation to a yield partition can then be obtained as

$$\begin{aligned} Y_i(\mathbf{x}_D) \approx \Theta[\beta_i(\mathbf{x}_D)] &= (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{\beta_i(\mathbf{x}_D)} \exp\left(\frac{-t^2}{2}\right) \mathrm{d}t \\ &= \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\beta_i(\mathbf{x}_D)}{\sqrt{2}}\right)\right]. \end{aligned} \tag{5}$$

Figure 1(a) illustrates this approximation. The true boundary of the acceptance region is replaced by a hyperplane (dashed line). The approximate yield partition $Y_i$ can then be calculated from $\beta_i$ using Eq. (5). The integral of the JPDF over



(a) (b)

Fig. 1. (a) The WCD problem illustrated for two statistical parameters. The contours of $\min_{\mathbf{x}_R \in \mathcal{R}} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S)$ for a fixed set of design parameters $\mathbf{x}_D$ are depicted by solid curves. The acceptance region of $f_i$ is shaded in dark gray. $\beta_i$ denotes the WCD. The hashed region results in the acceptance region approximation error when $Y_i$ is replaced by $\Theta[\beta_i]$ (Eq. (5)). (b) Two WPM problems, one for $\beta_i^W < \beta_i$ and one for $\beta_i^{W\prime} > \beta_i$. The solution ($f_i^W$ and $f_i^{W\prime}$, respectively) is denoted by a dot. The two solutions satisfy $f_i^W > F_i > f_i^{W\prime}$. See Fig. 1(a) for the explanation of contours and shading. If the solution of the WPM problem for $\beta_i^W$ satisfies $f_i^W \ge F_i$ then the yield partition satisfies $Y_i \ge \Theta[\beta_i^W]$ (within the acceptance region approximation error).

the hashed region in Fig. 1(a) results in the acceptance region approximation error and is usually small due to the nature of the normal distribution.[1] Maximizing the yield now becomes equivalent to maximizing the smallest yield partition, which in turn means that the WCD of the performance measure with the smallest $\beta_i$ must be maximized.

Instead of maximizing the smallest WCD we can take a different approach. $\beta(\mathbf{x}_S) \leq \beta_i^W$ defines a tolerance body in the space of statistical parameters for $i$th performance measure. Together with the tolerance body of the range parameters ($\mathcal{R}$) they define the search space in which we are looking for the smallest (worst) value $f_i^W(\mathbf{x}_D)$ of performance measure $f_i$ (WPM problem):

$$f_i^W(\mathbf{x}_D) = \min_{\mathbf{x}_R \in \mathcal{R}, \beta(\mathbf{x}_S) \leq \beta_i^W} f_i(\mathbf{x}_D, \mathbf{x}_R, \mathbf{x}_S)\,. \tag{6}$$

Note that in Eq. (6) $\beta_i^W$ no longer represents the WCD, but a user defined constraint corresponding to the $i$th performance measure. The following estimate holds (within acceptance region approximation error)

$$Y_i(\mathbf{x}_D) \geq \Theta[\beta_i^W]\,. \tag{7}$$

By maximizing $f_i^W(\mathbf{x}_D) - F_i$ we maximize the yield partition $Y_i(\mathbf{x}_D)$. If we do this for all performance measures simultaneously we maximize the total yield. Typically one would choose $\beta_i^W = 3$ which would result in $Y_i \geq 0.9987$ for a circuit satisfying $f_i^W \geq F_i$.

Figure 1(b) illustrates the WPM problem. $\beta_i^W$ and $\beta_i^{W\prime}$ satisfy $\beta_i^W \leq \beta_i \leq \beta_i^{W\prime}$. Due to this the corresponding solutions of the WPM problem satisfy $f_i^W \geq F_i \geq f_i^{W\prime}$. Many performance measures have no local minima within the search space, so the worst point is often located at $\beta(\mathbf{x}_S) = \beta_i^W$ (just as in Fig. 1(b)).

Because all constraints in the WPM problem are simple (e.g., box constraints on the range parameters and a spherical constraint on the statistical parameters), it can be solved by adapting a direct search method to the nature of these constraints.

### 2.3. *The optimization problem arising from the WPM problem*

Denote by $\mathbb{R}^n$ a $n$-dimensional Hilbert space and let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two orthogonal subspaces of $\mathbb{R}^n$ for which $\mathbb{R}^n = (\mathcal{S}_1 \oplus \mathcal{S}_2)$. Then every $\mathbf{x} \in \mathbb{R}^n$ can be uniquely expressed as $\mathbf{x} = \mathcal{P}_{\mathcal{S}_1}(\mathbf{x}) + \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})$ where $\mathcal{P}_{\mathcal{S}}(\mathbf{x})$ denotes the orthogonal projection of $\mathbf{x}$ on subspace $\mathcal{S}$. Let $\{\mathbf{e}_i : \|\mathbf{e}_i\| = 1,\ i = 1, \ldots, n_1\}$ and $\{\mathbf{b}_i : i = 1, \ldots, n_2\}$ denote an orthonormal linear basis for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Note that $n_1 + n_2 = n$. Then every $\mathbf{x} \in \mathbb{R}^n$ can be expressed with the basis vectors as

$$\mathbf{x} = \sum_{i=1}^{n_1} x_i \mathbf{e}_i + \sum_{i=1}^{n_2} x_{n_1+i} \mathbf{b}_i\,, \tag{8}$$

where $\sum_{i=1}^{n_1} x_i \mathbf{e}_i = \mathcal{P}_{\mathcal{S}_1}(\mathbf{x})$ and $\sum_{i=1}^{n_2} x_{n_1+i} \mathbf{b}_i = \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})$.

We are trying to solve the the following optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to}\,, \tag{9}$$

$$\|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})\| \le \beta^W\,, \tag{10}$$

$$x_{Li} \le x_i \le x_{Hi}\,, \quad i = n_1+1, n_1+2, \ldots, n_1+n_2\,, \tag{11}$$

where $x_{Li} < x_{Hi}$ and $\beta^W > 0$. Constraint Eq. (10) defines a closed ball in subspace $\mathcal{S}_1$ while the inequality constraints Eq. (11) are box constraints in subspace $\mathcal{S}_2$. The $i$th left (right) inequality in Eq. (11) defines the $i$th lower (upper) box constraint. Note that subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ correspond to the space of statistical and range parameters, respectively.

We try to solve problem Eqs. (9) to (11) using direct search. A prerequisite for the convergence of direct search methods to a local minimizer of a continuously differentiable function is that the search directions conform to the boundary of the search region. This can be achieved by using steps that in limit (as step size approaches zero) span the tangent space and the normal space.[21, 22]

This is straightforward for orthogonal box constraints in subspace $\mathcal{S}_2$. One simply uses all unit vectors that are normal to bounds as search steps. For subspace $\mathcal{S}_1$, directions that span the tangential plane of the hypersphere and two radial directions can be used. Using directions chosen in this manner a simple barrier approach can be used for handling the constraints.

## 3. The Proposed Algorithm for Solving the WPM Problem

### 3.1. *Moving on the surface of a hypersphere*

The steps that in limit span the tangential space of a hypersphere

$$\|\mathbf{x}\| = \beta^W\,; \quad \mathbf{x} \in \mathbb{R}^{n_1}\,, \tag{12}$$

in subspace $\mathcal{S}_1$ can be defined using rotations. Take for instance, point $\mathbf{x}$ on the surface of hypersphere Eq. (12). A rotation of vector $\mathbf{x}$ can be uniquely described by a two dimensional plane and the angle of rotation. The plane is defined by $\mathbf{x}$ and an additional vector $\mathbf{a} \neq \mathbf{0}$ where $\mathbf{x}$ and $\mathbf{a}$ are linearly independent. The rotation of vector $\mathbf{x}$ by an angle $\varphi$ toward $\mathbf{a}$ can then be expressed using vector $\mathbf{a}'$ that lies in the above mentioned two dimensional plane and is orthogonal to $\mathbf{x}$:

$$\mathbf{a}' = \mathbf{a} - \frac{\mathbf{a}^T\mathbf{x}}{\|\mathbf{x}\|^2}\mathbf{x}\,. \tag{13}$$

The rotated vector $\mathbf{x}'$ is obtained from

$$\mathbf{x}' = \operatorname{rotate}(\mathbf{x}, \mathbf{a}, \varphi) = \begin{cases} \mathbf{x}\cos\varphi + \dfrac{\mathbf{a}'}{\|\mathbf{a}'\|}\|\mathbf{x}\|\ \sin\varphi\,; & \mathbf{x} \text{ and } \mathbf{a} \text{ linearly independent}\,, \\ \mathbf{x}\,; & \text{otherwise}\,. \end{cases} \tag{14}$$

The case when $\mathbf{x}$ and $\mathbf{a}$ are not linearly independent is included for formal reasons and makes the description of the algorithm more concise. Effectively it means that in such case no rotation is performed. If the hypersphere is defined by Eq. (10), a step on the surface of the hyperspheres can be expressed as

$$\mathbf{x}' = \mathrm{rotate}_1(\mathbf{x}, \mathbf{a}, \varphi) = \mathrm{rotate}[\mathcal{P}_{\mathcal{S}_1}(\mathbf{x}), \mathcal{P}_{\mathcal{S}_1}(\mathbf{a}), \varphi] + \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})\,. \tag{15}$$

It is easy to prove that vector $\mathbf{x}'$ also represents a point on the hypersphere.

### 3.2. *Moving in radial direction*

Radial search directions span the normal space of constraint Eq. (12). We define a radial step as

$$\mathbf{x}^* = \mathrm{radial}(\mathbf{x}, \Delta) = \mathbf{x} + \frac{\mathbf{x}}{\|\mathbf{x}\|}\Delta\,, \tag{16}$$

where $\Delta$ is the step size. A problem can occur when $\|\mathbf{x}^*\| = 0$. For such point no rotations are performed. This weakness can be removed by avoiding steps that fall within $\beta_{\min}$ of $\mathbf{0}$. If a radial step in direction $\mathbf{x}/\|\mathbf{x}\| \cdot \Delta$ results in point $\mathbf{x}^*$ inside the hypersphere ($\|\mathbf{x}^*\| < \beta_{\min}$) then the step is modified so that the resulting point $\mathbf{x}^{**}$ lies on the surface. It can be expressed as:

$$\mathbf{x}^{**} = -\frac{\mathbf{x}}{\|\mathbf{x}\|}\beta_{\min}\,. \tag{17}$$

If the hypersphere is defined by Eq. (10), a radial step together with Eq. (17) can be expressed as

$$\begin{aligned}\mathbf{x}^* &= \mathrm{radial}_1(\mathbf{x}, \Delta, \beta_{\min})\\ &= \begin{cases}\mathrm{radial}[\mathcal{P}_{\mathcal{S}_1}(\mathbf{x}), \Delta] + \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})\,; & \|\mathrm{radial}[\mathcal{P}_{\mathcal{S}_1}(\mathbf{x}), \Delta]\| \geq \beta_{\min}\,,\\ -\dfrac{\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})}{\|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})\|}\beta_{\min} + \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})\,; & \text{otherwise}\,.\end{cases}\end{aligned} \tag{18}$$

### 3.3. *The proposed algorithm*

The idea for the proposed algorithm presented in this section comes from the Hooke–Jeeves algorithm which consists of two parts.[23] The first part uses trial steps and searches for a lower value of the function $f$ in the neighborhood of the current point. If a better point is found, the second part performs a speculative step which is supposed to speed up the search. After a speculative step is taken, the function is not evaluated. The algorithm takes a trial step superimposed to the speculative step and if it produces descent with respect to the last successful trial step, the speculative step is considered successful. If the trial step fails, the speculative step is considered as failed and the algorithm returns to the last successful trial step. A detailed description of trial and speculative steps can be found in the original paper.[23]

Suppose we are trying to solve problem Eqs. (9) to (11). The proposed algorithm takes constrains (Eqs. (10) and (11)) into account by performing trial and speculative steps that conform to the boundary of the search region. If constrains are violated, the barrier approach is used. This is the first of the two main differences between the proposed algorithm and the original unconstrained Hooke–Jeeves algorithm. The second main difference is that the proposed algorithm searches in two subspaces simultaneously use different trial and speculative steps for every subspace. For this purpose two new kinds of steps are introduced: rotations and radial steps.

Let us define a few things in order to simplify the description of the proposed algorithm. Constraints Eqs. (10) and (11) can be violated when the algorithm makes a step. Function $\Omega_1(\mathbf{x})$:

$$\Omega_1(\mathbf{x}) = \begin{cases} \mathbf{x}\,; & \|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})\| \le \beta^W\,, \\ \dfrac{\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})}{\|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})\|}\beta^W + \mathcal{P}_{\mathcal{S}_2}(\mathbf{x})\,; & \|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})\| > \beta^W\,, \end{cases} \tag{19}$$

makes sure constraint Eq. (10) is not violated by moving points that violate the constraint to the boundary of the search region. A similar function can be defined for enforcing the constraints Eq. (11) in subspace $\mathcal{S}_2$:

$$\Omega_2(\mathbf{x}) = \begin{cases} x_i\,; & 1 \le i \le n_1\,, \\ x_i\,; & n_1+1 \le i \le n_1+n_2 \ \wedge \ x_{Li} \le x_i \le x_{Hi}\,, \\ x_{Li}\,; & n_1+1 \le i \le n_1+n_2 \ \wedge \ x_i < x_{Li}\,, \\ x_{Hi}\,; & n_1+1 \le i \le n_1+n_2 \ \wedge \ x_i > x_{Hi}\,. \end{cases} \tag{20}$$

We denote the composite of functions $\Omega_1$ and $\Omega_2$ by

$$\Omega(\mathbf{x}) = \Omega_2(\Omega_1(\mathbf{x}))\,. \tag{21}$$

Function

$$\mathbf{P}_T(\mathbf{x}, \Delta, i) = \begin{cases} \text{rotate}_1[\mathbf{x}, \mathbf{e}_i, \Delta_i]\,; & 1 \le i \le n_1\,, \\ \text{radial}_1[\mathbf{x}, \Delta_i, \beta_{\min}]\,; & i = n_1+1\,, \\ \mathbf{x} + \Delta_i b_{i-n_1-1}\,; & n_1+2 \le i \le n_1+n_2+1\,, \end{cases} \tag{22}$$

produces a trial step across subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. The first type of trial steps in subspace $\mathcal{S}_1$ are rotations of $\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})$ toward some basis vector $\mathbf{e}_i$. If $\mathbf{e}_i$ is collinear with $\mathcal{P}_{\mathcal{S}_1}(\mathbf{x})$ the corresponding trial step is omitted (which is handled in Eq. (14) by omitting the rotation). The second type of trial steps in subspace $\mathcal{S}_1$ are radial steps. Trial steps in subspace $\mathcal{S}_2$ are the same as those in the original Hooke–Jeeves algorithm. The step size $\Delta$ is given by $\varphi$ and $R$ for subspace $\mathcal{S}_1$ and $\Phi$ for subspaces $\mathcal{S}_2$:

$$\Delta_i = \begin{cases} \varphi\,; & 1 \le i \le n_1\,, \\ R\,; & i = n_1+1\,, \\ \Phi\,; & n_1+2 \le i \le n_1+n_2+1\,. \end{cases} \tag{23}$$

Radial trial steps in subspace $\mathcal{S}_1$ and trial steps in subspace $\mathcal{S}_2$ are considered successful if they result in simple descent (i.e., the function value at the new point is lower than at the old point). A more strict requirement is imposed on trial steps that are rotations in subspace $\mathcal{S}_1$. They must produce sufficient descent. The amount of descent that is considered sufficient is prescribed by $\gamma$. This is one of the requirements that make sure the algorithm is convergent.[21]

$$d_i = \begin{cases} \gamma\,; & 1 \le i \le n_1\,, \\ 0\,; & i = n_1 + 1\,, \\ 0\,; & n_1 + 2 \le i \le n_1 + n_2 + 1\,. \end{cases} \tag{24}$$

$\angle_{TB}$ denotes the angle between the projections of $\mathbf{x}_B$ and $\mathbf{x}_T$ to subspace $\mathcal{S}_1$ (see proposed algorithm, label 1). Similarly, $D_{TB}$ represents $\|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x}_T)\| - \|\mathcal{P}_{\mathcal{S}_1}(\mathbf{x}_B)\|$. The proposed algorithm for solving the WPM problem can now be stated.

```
    Choose $\mathbf{x}_0$ satisfying Eqs. (10) and (11).
    Choose $\varphi > 0$, $R > 0$, $\Phi > 0$, $\gamma > 0$, $0 < \beta_{\min} < \beta^W$, $r > 1$, $\delta > 1$ and $\alpha > 1$.
    $\mathbf{x}_B := \mathbf{x}_0$, $f_B := f(\mathbf{x}_B)$, $\mathbf{x}_T := \mathbf{x}_B$, and $f_T := f_B$
    $sp := 0$;
    while not stopping condition satisfied do
        if $f_T < f_B$ then
            // attempt speculative step
            $sp := 1$;
1:          $\mathbf{x}_{SP} := \mathrm{rotate}_1\,[\mathbf{x}_B, \mathbf{x}_T, \alpha\angle_{TB}]$
            $\mathbf{x}_{SP} := \mathrm{radial}_1[\mathbf{x}_{SP}, \alpha D_{TB}, \beta_{\min}]$;
            $\mathbf{x}_{SP} := \mathbf{x}_{SP} + \alpha\mathcal{P}_{\mathcal{S}_2}(\mathbf{x}_T - \mathbf{x}_{SP})$;
            $\mathbf{x}_B := \mathbf{x}_T$;  $f_B := f_T$;
            $\mathbf{x}_T := \Omega(\mathbf{x}_{SP})$;
        else
            // speculative step failed
            $sp := 0$;
            $\mathbf{x}_T := \mathbf{x}_B$;  $f_T := f_B$;
        end
2:      for $(i := 1, n_1 + 2, 2, 3, \ldots, n_1 + 1, n_1 + 3, n_1 + 4, \ldots, n_1 + n_2 + 1)$ do
            // trial step (+)
            $\mathbf{x}_{Ttemp} := \Omega(\mathbf{P}_T(\mathbf{x}_T, +\Delta, i))$;
            if $f(\mathbf{x}_{Ttemp}) < f_T - d_i$ then
                // trial step (+) accepted
                $\mathbf{x}_T := \mathbf{x}_{Ttemp}$; $f_T := f(\mathbf{x}_{Ttemp})$;
            end
3:          if $f(\mathbf{x}_{Ttemp}) \ge f_T - d_i$ or $sp = 1$ then
                // trial step (-)
                $\mathbf{x}_{Ttemp} := \Omega(\mathbf{P}_T(\mathbf{x}_T, -\Delta, i))$;
                if $f(\mathbf{x}_{Ttemp}) < f_T - d_i$ then
                    // trial step (-) accepted
                    $\mathbf{x}_T := \mathbf{x}_{Ttemp}$;  $f_T := f(\mathbf{x}_{Ttemp})$;
                end
            end
```

```
4:        if $\mathrm{sp} = 1$ and $f_T \geq f_B$ and $i = n_1 + 2$ then
             break;
          end
       end
       if $sp = 0$ and $f_T \geq f_B$ then
          // all trial steps failed
          $\varphi := \varphi/r$; $R = R/r$; $\Phi := \Phi/r$;
          $\gamma := \gamma/(r^{\delta})$; $\beta_{\min} := \beta_{\min}/r$;
       end
       if $\Phi < \pi/36$ then
          // stopping condition is satisfied
          break;
       end
    end
```

The step acceptance criterion for rotations is more strict than in the Hooke–Jeeves algorithm. We require sufficient descent instead of simple descent. After a speculative step is taken, the proposed algorithm performs trial steps in both directions (+ and −) and accepts the one that results in greater reduction of $f$ (proposed algorithm, label 3). This helps avoid the situation when a speculative step in one direction is followed by an accepted trial step in the opposite direction, which reduces the effect of the speculative step. The fact that some trial points are identical to $\mathbf{x}_T$ because $\Omega$ is applied to the result of $\mathbf{P}_T$, helps reduce the number of function evaluations. If a trial point is identical to $\mathbf{x}_T$, the trial step can be considered as failed without evaluating $f$ because such steps result in no descent.

The **for** statement (proposed algorithm, label 2) has a nonstandard trial step series. The reason is that the speculative step is accepted if the first trial step in subspace $\mathcal{S}_1$ (corresponding to $i = 1$) or subspace $\mathcal{S}_2$ (corresponding to $i = n_1 + 2$) produces descent.

The proposed algorithm is also suitable for optimization problems where either $n_2 = 0$ or $n_1 = 0$. In this case the indices ($i$) traversed by the **for** statement (proposed algorithm, label 2) are $i = 1, \ldots, n_1 + 1$ or $i = 1, \ldots, n_2$, respectively. In this case, the speculative step is followed by the first trial step in only one subspace. Therefore the condition of the last **if** statement (proposed algorithm, label 4) changes from $i = n_1 + 2$ to $i = 1$.

The algorithm can be proved to converge to a first-order constrained local optimum of a continuously differentiable function (performance measure) using known techniques.[21]

## 4. Examples

Analog integrated circuits presented in this section provide vital functions to many analog and mixed signal circuits. For the statistical parameters we limit ourselves to two electrical parameters of every MOS transistor: threshold voltage ($V_T$) and current factor ($K$). In subsequent examples the Pelgrom model for transistor mismatch is used[16]:

$$\sigma(\Delta V_T) = \frac{A_{V_T}}{\sqrt{WL}}\,, \tag{25}$$

$$\sigma\left(\frac{\Delta K}{K}\right) = \frac{A_K}{\sqrt{WL}}\,. \tag{26}$$

In this model the difference between the values of $V_T$ and $K$ of two identically designed transistors with width $W$ and length $L$ (note that $W$ and $L$ are design parameters) is modeled by a normal JPDF with zero mean, zero correlation, and standard deviations that depend on transistor dimensions. The threshold voltage and the current factor of one particular transistor can then be expressed as

$$V_T = \mu_{V_T} + N_{V_T}(0,1)\frac{A_{V_T}}{\sqrt{2WL}}\,, \tag{27}$$

$$K = \mu_K\left[1 + N_K(0,1)\frac{A_K}{\sqrt{2WL}}\right], \tag{28}$$

where $\mu_{V_T}$ and $\mu_K$ represent the mean value of threshold voltage and current factor, and $N_{V_T}(0,1)$ and $N_K(0,1)$ denote two uncorrelated normal random variables with zero mean ($\mathbf{x}_S^N = 0$) and standard deviation 1. If we choose these normal random variables as our statistical parameters, we have a total of $2M$ statistical parameters where $M$ is the number of transistors in the circuit. Furthermore, $\beta(\mathbf{x}_S) = \beta_i^W$ defines a sphere and the proposed algorithm can be directly applied to the calculation of the worst performance measure value $f_i^W$. A more advanced mismatch model can be used by changing the formulas that define the variances and covariances of statistical parameters and applying a linear transformation so that the spherical shape of the JPDF is restored.[24]

In the subsequent examples, worst-case values of performance measures for several circuits have been calculated. All circuits in this paper have been simulated using the SPICE OPUS simulator and the BSIM3 model of a 0.18 $\mu$m process technology (predictive technology model).[25–27] The mismatch parameters $A_{V_T}$ and $A_K$ were obtained from Kinget's paper.[28]

The computation of the worst-case value of a performance measure $f_i^W$ can be divided in three major steps. In the first step, statistical parameters $\mathbf{x}_S$ are kept unchanged at 0 and only range parameters $\mathbf{x}_R$ are subject to optimization. First, the performance is measured at the maximum and the minimum value of each range parameter, while the remaining range parameters are left at their nominal values. Based on the results, the extreme values of range parameters are combined in a manner that would yield the worst performance measure value if the performance was a linear function of the range parameters. The resulting set of extreme range parameters is used as the starting point for the proposed algorithm. If the performance is not a linear function of range parameters, then such initial point may not be ideal. In this case, the proposed algorithm may require additional performance evaluations. Our experience shows that the proposed approach reduces the number of performance evaluations for most real-world cases. In this step, statistical parameters are not subject to optimization and are kept at their nominal value $\mathbf{x}_S^N = \mathbf{0}$. The algorithm results in a range parameter vector $(\mathbf{x}_R^{W_0})$. This step often speeds up the remainder of the procedure because worst range parameter values are usually independent of statistical parameters.

In the second step, the initial sufficient descent $\gamma$ and the initial gradient direction with respect to the statistical parameters are calculated. For every statistical parameter $x_{Sj}$ (where $j = 1, \ldots, n_S$) two circuit simulations are performed, one for $x_{Sj} = +\beta_i^W$ and one for $x_{Sj} = -\beta_i^W$ resulting in $f_i^{+j}$ and $f_i^{-j}$. All other statistical parameters are set to 0 and range parameters are set to $\mathbf{x}_R^{W_0}$. In total we have $2n_S$ circuit simulations with $2n_S$ values of the $i$th performance measure. Let $f_i^{\max}$ and $f_i^{\min}$ denote the largest and the smallest value of $f_i$ obtained in these $2n_S$ simulations. The initial sufficient descent is then obtained as

$$\gamma = \frac{(f_i^{\max} - f_i^{\min})}{10} . \tag{29}$$

Pairs of values $f_i^{+j}$ and $f_i^{-j}$ (where $j = 1, \ldots, n_S$) enable us to estimate the gradient $\nabla f_i$ of $f_i$ with the central difference formula. This gradient helps us choose an initial guess of the worst statistical parameters for the third step where the main part of the $f_i^W$ computation takes place. In this step, both range and statistical parameters are subject to optimization. The initial values are set to $\mathbf{x}_R = \mathbf{x}_R^{W_0}$ and $\mathbf{x}_S = \beta_i^W \nabla f_i / \|\nabla f_i\|$. The algorithm results in the worst performance measure value $f_i^W$ and the corresponding $\mathbf{x}_R^W$ and $\mathbf{x}_S^W$. Throughout all examples we have chosen $\beta_i^W = 3$, which corresponds to a 99.87% yield partition.

One may argue that using a local optimization method like the one we are proposing would not yield the true WPM value because it is prone to getting trapped in local minima (local worst performance). But if we consider that current state-of-the-art method in WCD computation also uses a local search method (sequential quadratic programming — SQP) and produces good results on real-world circuits,[7,29] we conclude that the performance measures of most circuits have few local minima within the commonly used tolerance bodies of statistical and range parameters.

The initial step size for the statistical parameters was set to $\varphi = \pi/4$ and $R = \beta_i^W/2$ (third step of the algorithm). The initial value of $\beta_{\min}$ was set to $\beta_i^W/3$. Range parameters were normalized in such manner that full range corresponded to the $[-\pi, \pi]$ interval. The initial step for the range parameters was $\Phi = \pi/4$. The value of $r$, $\alpha$ and $\delta$ was set to 6, 2, and 2, respectively.

### 4.1. *Performance of the proposed algorithm*

All examples have two range parameters in common: temperature ranging from −20°C to 80°C and supply voltage ranging from 1.6 V to 2.0 V. The simple

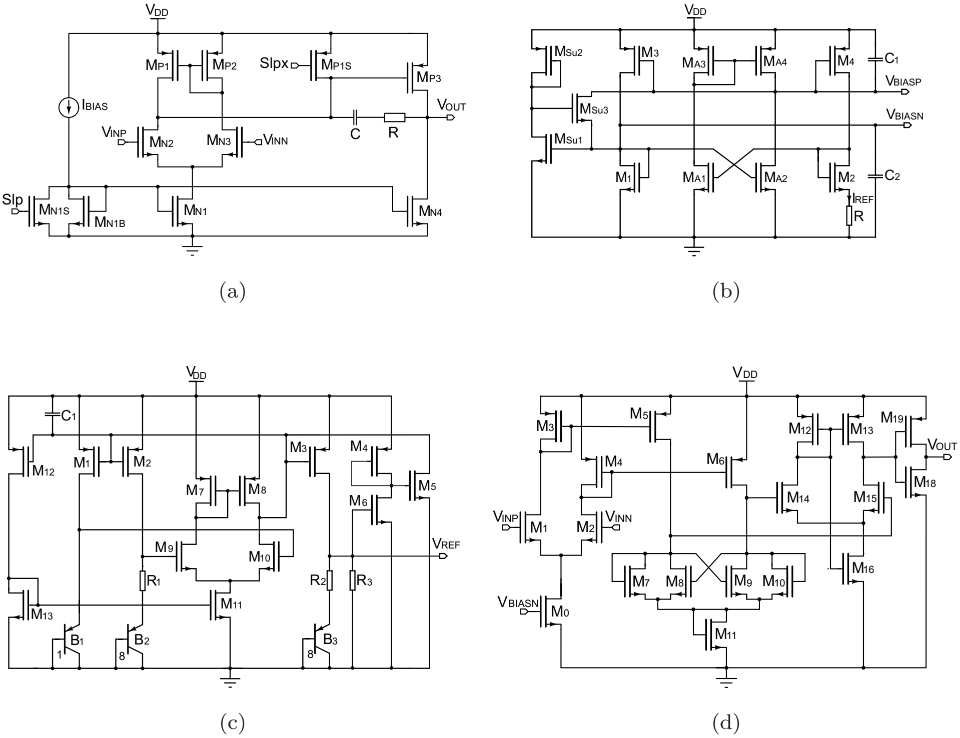


Fig. 2. The test circuits: (a) a simple operational amplifier; (b) a beta multiplier reference; (c) a bandgap reference; and (d) a comparator.

operational amplifier case has an additional range parameter — the bias current ranging from $80\,\mu$A to $120\,\mu$A.

We test the performance of the proposed algorithm on four test circuits depicted in Fig. 2. For the simple operational amplifier (OPAMP) the swing at gain is measured as the range of output voltage for which the differential gain is above 50% of the differential gain at zero input. The offset voltage is expressed as the upper and the lower bound on the output voltage variation at zero input voltage arising from statistical parameter variations. Each transistor of the OPAMP contributes 2 statistical parameters except for transistors $M_{N1S}$ and $M_{P1S}$ that are used for shutting down the amplifier and thus have no influence on its performance.[30]

For the beta multiplier reference (BMR) circuit, the current that flows through resistance $R$ is copied to other branches of a circuit that utilizes the BMR using gate voltages $V_{BIASP}$ and $V_{BIASN}$. Ideally the reference current should exhibit no temperature dependence nor supply voltage dependence. This is, of course not true in practice therefore we observe its variation with respect to the two range parameters. In this circuit there are 16 statistical parameters ($M_{SU1}$, $M_{SU2}$, and $M_{SU3}$ compose the start-up circuit and result in no additional statistical parameters).[30]

The bandgap reference (BGR) circuit is a low power CMOS circuit based on resistive subdivision.[31] It has 20 statistical parameters. Transistors $M_4$, $M_5$, and $M_6$ comprise the start up circuit and contribute no statistical parameters. In results we list the same performance measures as for the BMR circuit.

In the comparator circuit (Fig. 2(d)) variations of statistical parameters cause a significant increase of hysteresis. The BMR reference circuit (see Fig. 2(b)) was used as a source of reference current for the comparator. The complete comparator has 54 statistical parameters (16 due to the BMR circuit and 38 due to the comparator).[30]

The results of the worst case analysis are listed in Table 1. An upward (downward) pointing arrow indicates that one desires the measure to be as high (low) as possible. The columns list the performances measures ($f_i$) and their values at nominal statistical ($\mathbf{x}_S^N$) and nominal range ($\mathbf{x}_R^N$) parameters ($f^N$), the worst $f_i$ across range parameters (i.e., at $\mathbf{x}_R^{W_0}$ and $\mathbf{x}_S^N$) obtained in the first step ($f^{WN}$), and the worst $f_i$ across both range and statistical parameters ($f^W$) (obtained in the third step). The numbers in parenthesis give the number of performance measure evaluations (simulations) in the first step (the $f^{WN}$ column) and second and third step (the $f^W$ column).

The fifth and sixth columns lists the worst value ($f_{MC}$) of 500,000 Monte-Carlo simulations and the results of the WPM algorithm ($f_{MC}^W$) when the initial point was set to the worst case obtained by the Monte-Carlo analysis. The range and statistical parameters of Monte-Carlo samples were randomly chosen from the box (Eq. (11)) and the closed ball ($\beta_i^W \leq 3$). The comparison of the Monte-Carlo-based results ($f_{MC}$ and $f_{MC}^W$) with the results obtained by the proposed algorithm ($f^W$) indicates that the proposed method is a reliable way for finding the worst performance measure. The proposed algorithm found the same or even better result than $f_{MC}^W$ in all cases.

The proposed algorithm's performance is also listed in Table 1. The numbers in parentheses and the *iter* column list the number of performance measure evaluations (effectively this is the number of circuit simulations). In the last column, the equivalent number of performance measure gradient evaluations is listed. We assume that the central difference formula is used for gradient evaluation. As it can be seen from the table the WPM evaluation using the proposed algorithm in most cases corresponds to 3–7 gradient evaluations. This indicates that the proposed optimization algorithm is an efficient approach for solving the WPM problem.

In a paper by Antreich *et al.*, the number of modified SQP iterations for solving the WCD problem is between 5 and 9.[7,32] Note that one SQP iteration involves one gradient evaluation. Beside the gradient evaluation, the line search and the second-order correction also contribute to the number of performance evaluations. By taking this into account, we can conclude that the proposed algorithm for solving the WPM problem has similar performance as the state-of-the-art algorithm for solving the WCD problem.

Table 1. Results of WPM algorithm and equivalent number of gradient evaluations.

| Performance name | | $f^N$ | $f^{WN}$ | $f^W$ | $f_{MC}$ | $f^W_{MC}$ | $iter$ | $\frac{iter}{2(n_S+n_R)}$ |
|---|---|---|---|---|---|---|---|---|
| *Operational amplifier*, $n_S = 16$, $n_R = 3$ | | | | | | | | |
| Slew rate rise $[\frac{\mathrm{V}}{\mu\mathrm{s}}]$ | ↑ | 42 | 30 (13) | 29 (105) | 30 | 29 | 118 | 3.1 |
| Slew rate fall $[\frac{\mathrm{V}}{\mu\mathrm{s}}]$ | ↑ | 47 | 37 (30) | 37 (105) | 37 | 37 | 135 | 3.6 |
| Settling time rise [$\mu$s] | ↓ | 0.52 | 0.54 (13) | 0.54 (105) | 0.54 | 0.54 | 118 | 3.1 |
| Settling time fall [$\mu$s] | ↓ | 0.53 | 0.55 (39) | 0.55 (105) | 0.55 | 0.55 | 144 | 3.8 |
| Delay rise [ns] | ↓ | 4.9 | 8.0 (13) | 8.3 (105) | 8.0 | 8.3 | 118 | 3.1 |
| Overshoot [%] | ↓ | 7.1 | 10.9 (25) | 11.4 (187) | 11.1 | 11.4 | 212 | 5.6 |
| Undershoot [%] | ↓ | 6.7 | 11.4 (35) | 12.0 (147) | 11.6 | 12.0 | 182 | 4.8 |
| Swing at gain [V] | ↑ | 1.37 | 1.13 (13) | 1.13 (258) | 1.13 | 1.13 | 271 | 7.1 |
| Phase margin [°] | ↑ | 59.4 | 54.4 (34) | 54.0 (185) | 54.3 | 54.0 | 219 | 5.8 |
| Unity gain BW [MHz] | ↑ | 99.5 | 55.2 (13) | 52.7 (105) | 55.3 | 52.8 | 118 | 3.1 |
| Gain at 0 Hz [dB] | ↑ | 60.0 | 58.6 (13) | 58.4 (105) | 58.5 | 58.4 | 118 | 3.1 |
| Out offset (upper) [mV] | ↓ | 0 | 0 (27) | 36.1 (227) | 30.0 | 35.9 | 254 | 6.7 |
| Out offset (lower) [mV] | ↑ | 0 | 0 (30) | −36.0 (307) | −30.1 | −35.7 | 337 | 8.9 |
| *Beta-multiplier reference*, $n_S = 16$, $n_R = 2$ | | | | | | | | |
| $\frac{\Delta I_{REF}}{\Delta T}$ $[\frac{\mathrm{nA}}{^\circ\mathrm{C}}]$ | ↓ | 61 | 85 (9) | 93 (103) | 90 | 92 | 112 | 3.1 |
| $\frac{\Delta I_{REF}}{\Delta V_{DD}}$ $[\frac{\mathrm{nA}}{\mathrm{V}}]$ | ↓ | 14 | 19 (9) | 31 (103) | 27 | 31 | 112 | 3.1 |
| *Bandgap voltage reference*, $n_S = 20$, $n_R = 2$ | | | | | | | | |
| $\frac{\Delta V_{REF}}{\Delta T}$ $[\frac{\mu\mathrm{V}}{^\circ\mathrm{C}}]$ | ↓ | 50 | 50 (27) | 230 (503) | 162 | 225 | 530 | 12.0 |
| $\frac{\Delta V_{REF}}{\Delta V_{DD}}$ $[\frac{\mathrm{mV}}{\mathrm{V}}]$ | ↓ | 3.8 | 8.1 (9) | 63 (173) | 42 | 62 | 182 | 4.1 |
| *Comparator*, $n_S = 54$, $n_R = 2$ | | | | | | | | |
| Output rise time [ns] | ↓ | 1.0 | 1.3 (15) | 1.3 (565) | 1.3 | 1.3 | 580 | 5.2 |
| Output fall time [ns] | ↓ | 1.4 | 1.8 (9) | 1.8 (559) | 1.8 | 1.8 | 568 | 5.1 |
| Output rise delay [ns] | ↓ | 10 | 12 (9) | 12 (445) | 12 | 12 | 454 | 4.1 |
| Output fall delay [ns] | ↓ | 9.7 | 10 (9) | 11 (445) | 11 | 11 | 454 | 4.1 |
| Hysteresis [mV] | ↓ | 3.1 | 5.1 (9) | 9.2 (331) | 7.1 | 9.1 | 340 | 3.0 |

### 4.2. *An example of yield maximization*

The proposed WPM algorithm can be used for obtaining a robust design with a high yield. We demonstrate the approach on the operational amplifier example. The results are in Table 2. The initial circuit has good nominal performances ($f_{ini}^N$) but many of its worst performance measures $f_{ini}^W$ fail to satisfy the design requirements (goal) which is the cause of low yield ($Y \approx 0\%$). To maximize the yield, the presented WPM algorithm was included in an optimization loop where the goal was to make all WPM values better than the corresponding goals. We chose $\beta_i^W = 3$ which corresponds to a 99.87% yield.

In the yield maximization algorithm, the proposed WPM algorithm and a standard parameter optimization algorithm are combined. First, the WPM algorithm is used to find a set of worst range and statical parameters for every performance measure. Next, the design parameters of the circuit are optimized so that all performance measures satisfy the respective goals for the corresponding worst range and statistical parameter values obtained by the WPM algorithm. The resulting circuit is subject to another WPM algorithm run resulting in new worst range and statistical parameter values which are added to the set of worst parameters. The optimization of design parameters is restarted with the updated set of worst range and statistical parameters. The two algorithms take turns until the worst performance obtained by the WPM algorithm satisfies all design requirements.

After the optimization is finished, the worst performance of the circuit ($f_{opt}^W$) satisfies all design requirements. This means that the yield of such a circuit should be above 99.87%. We verify this by performing a 10,000 sample Monte-Carlo analysis on the resulting circuit. The performance of every Monte-Carlo sample is taken at the corresponding worst range parameter values obtained by a WPM analysis. The Monte-Carlo analysis resulted in a 99.65% yield. Note that the obtained yield

Table 2. The performance of the operational amplifier before and after yield maximization.

| Measure name | Goal | $f_{ini}^N$ | $f_{ini}^W$ | $f_{opt}^W$ |
|---|---|---|---|---|
| Slew rate rise $[\frac{\text{V}}{\mu\text{s}}]$ | $> 4$ | 42 | 29 | 8.6 |
| Slew rate fall $[\frac{\text{V}}{\mu\text{s}}]$ | $> 4$ | 47 | 37 | 8.1 |
| Settling time rise [$\mu$s] | $< 1.0$ | 0.52 | 0.54 | 0.63 |
| Settling time fall [$\mu$s] | $< 1.0$ | 0.53 | 0.55 | 0.62 |
| Delay rise [ns] | $< 150$ | 4.9 | 8.3 | 34 |
| Overshoot [%] | $< 10$ | 7.1 | 11.4 | 9.8 |
| Undershoot [%] | $< 10$ | 6.7 | 12.0 | 5.2 |
| Swing at gain [V] | $> 1.0$ | 1.37 | 1.13 | 1.44 |
| Phase margin [°] | $> 55$ | 59.4 | 54.0 | 78.8 |
| Unity gain BW [MHz] | $> 8$ | 99.5 | 52.7 | 8.0 |
| Gain at 0 Hz [dB] | $> 60$ | 60.0 | 58.4 | 60.3 |
| Out offset (upper) [mV] | $< 15$ | 0 | 36.1 | 15.0 |
| Out offset (lower) [mV] | $> -15$ | 0 | $-36.0$ | $-14.9$ |

has an error of ±0.19% due to a limit of 10,000 samples in the Monte-Carlo analysis. The whole process (WPM analysis and optimization combined) took 22.8 hours (115,698 circuit simulations) on a 3.0 GHz computer. The approach can be parallelized to great extent, most notably the performance of a circuit with given design parameters at different sets of range and statistical parameters can be calculated in parallel. Because there is at least one such set for every performance measure we expect speedups proportional to the number of performance measures subject to optimization.

## 5. Conclusion

Obtaining a high yield is one of the key points of IC design. The automation of the design for yield process relies on algorithms for worst case evaluation, which in turn solve a particular optimization problem. Because design for yield is also an optimization problem, we have two nested optimizations. This requires the inner optimization algorithm to be as fast as possible. Because most circuit simulators do not evaluate the gradients of the performance measures, the most natural way to approach such problems is a direct search.

The worst case of a circuit can be expressed in several ways. The worst performance measure (WPM) approach is well suited for direct search. It results in a constrained optimization problem with bound constraints and an inequality constraint. We proposed a direct search algorithm for solving this problem. The algorithm draws its inspiration from the Hooke–Jeeves unconstrained direct search algorithm. The constraints are enforced by using the barrier approach and a set of search directions that conform to the constraints. The convergence of the algorithm to a first-order constrained local optimum of a continuously differentiable function can be proved using established techniques.

The proposed algorithm's performance was demonstrated on several IC designs. The proposed WPM algorithm finds the real (global) worst-case of a performance measure in all cases, despite the fact that it is a local search method. The computational complexity of the algorithm was measured by the number of performance measure evaluations needed for obtaining the WPM value. In most cases the algorithm took the equivalent of 3–7 finite difference gradient evaluations to obtain one WPM value. This indicates that the proposed algorithm solves the WPM problem in an efficient manner and is a good candidate for inclusion in an automated design for yield algorithm. The comparison of the equivalent number of gradient evaluations to the state-of-the-art algorithm for solving the WCD problem (which in turn is the complement of the WPM problem) is another confirmation of the algorithm's efficiency.

The proposed WPM algorithm was also included in a performance optimization loop where it was calculating sets of range and statistical parameters at which the circuit exhibits its worst performance. The obtained circuit's WPM values satisfied all design requirements resulting in a 99.65% yield.

## Acknowledgments

The research has been supported by the Ministry of Higher Education, Science and Technology of Republic of Slovenia within program P2-0246 — Algorithms and optimization methods in telecommunications.

## References

1. H. E. Graeb, *Analog Design Centering and Sizing* (Springer, Dordrecht, 2007), p. 195.
2. X. Li, P. Gopalakrishnan, Y. Xu and L. T. Pileggi, Robust analog/RF circuit design with projection-based performance modeling, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26** (2007) 2–15.
3. A. Dharchoudhury and S. M. Kang, Worst-case analysis and optimization of VLSI circuit performances, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **14** (1995) 481–492.
4. M. Sengupta, S. Saxena, L. Daldoss, G. Kramer, S. Minehane and J. Cheng, Application-specific worst case corners using response surfaces and statistical models, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **24** (2005) 1372–1380.
5. R. Storn, System design by constraint adaptation and differential evolution, *IEEE Trans. Evol. Comput.* **3** (1999) 22–34.
6. M. Jing, Y. Hao, J. Zhang and P. Ma, Efficient parametric yield optimization of VLSI circuit by uniform design sampling method, *Microelectron. Reliab.* **45** (2005) 155–162.
7. K. J. Antreich, H. E. Graeb and C. U. Wieser, Circuit analysis and optimization driven by worst-case distance, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **13** (1994) 57–71.
8. F. Filicori and V. A. Monaco, Design centering and tolerance assignment problems, *Electron. Lett.* **12** (1976) 507–509.
9. K. S. Tahim and R. Spence, A radial exploration approach to manufacturing yield estimation and design centering, *IEEE Trans. Circuits Syst.* **26** (1979) 768–774.
10. R. M. Lewis, V. Torczon and M. W. Trosset, Direct search methods: Then and now, *J. Comput. Appl. Math.* **124** (2000) 191–207.
11. R. M. Lewis and V. Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, *SIAM J. Optim.* **12** (2002) 1075–1089.
12. T. G. Kolda, R. M. Lewis and V. Torczon, A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints, Technical Report, SAND2006-5315, Sandia National Laboratories, Albuquerque, NM and Livermore (2006).
13. Á. Bűrmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar and T. Tuma, Automated robust design and optimization of integrated circuits by means of penalty functions, *AEÜ Int. J. Electron. Comm.* **57** (2003) 47–56.
14. T. Mukherjee, L. R. Carley and R. A. Rutenbar, Efficient handling of operating range and manufacturing line variations in analog cell synthesis, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **19** (2000) 825–839.
15. K. R. Lakshmikumar, R. A. Hadaway and M. A. Copeland, Characterization and modeling of mismatch in MOS transistor for precision analog design, *IEEE J. Solid-State Circuits* **21** (1986) 1057–1066.
16. M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, Matching properties of MOS transistors, *IEEE J. Solid-State Circuits* **24** (1989) 1433–1440.

17. C. Michael and M. Ismail, Statistical modeling of device mismatch for analog MOS integrated circuits, *IEEE J. Solid-State Circuits* **27** (1992) 154–166.
18. T. Serrano-Gotarredona and B. Linares-Barranco, Systematic width-and-length dependent CMOS transistor mismatch characterization and simulation, *Anal. Integr. Circuits Signal Process.* **21** (1999) 271–296.
19. M. F. Lan, A. Tammineedi and R. Geiger, Current mirror layout strategies for enhancing matching performance, *Anal. Integr. Circuits Signal Process.* **28** (2001) 9–26.
20. C. He, X. Dai, H. Xing and D. Chen, New layout strategies with improved matching performances, *Anal. Integr. Circuits Signal Process.* **49** (2006) 281–289.
21. T. G. Kolda, R. M. Lewis and V. Torczon, Stationarity results for generating set search for linearly constrained optimization, *SIAM J. Optim.* **17** (2006) 943–968.
22. D. W. Dreisigmeyer, Equality constraints, Riemannian manifolds and direct search methods, Los Alamos National Laboratory (2007).
23. R. Hooke and T. Jeeves, Direct search solutions of numerical and statistical problems, *J. ACM* **8** (1961) 212–229.
24. J. Bastos, M. Steyaert, A. Pergoot and W. Sansen, Mismatch characterization of submicron MOS transistors, *Anal. Integr. Circuits Signal Process.* **12** (1997) 95–106.
25. J. Puhan and T. Tuma, Optimization of analog circuits with SPICE3f4, *Proc. European Conf. Circuit Theory and Design*, Budapest, Hungary, August 1997, pp. 177–180.
26. Y. Cao, T. Sato, D. Sylvester, M. Orshansky and C. Hu, New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation, *Proc. Custom Integrated Circuits Conf.*, Orlando, USA, May 2000, pp. 201–204.
27. W. Zhao and Y. Cao, New generation of predictive technology model for sub-45nm early design exploration, *IEEE Trans. Electron. Dev.* **53** (2006) 2816–2823.
28. P. R. Kinget, Device mismatch and tradeoffs in the design of analog circuits, *IEEE J. Solid-State Circuits* **40** (2005) 1212–1224.
29. J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer Science+Business Media, New York, 2006), p. 664.
30. R. J. Baker, *CMOS Circuit Design, Layout, and Simulation* (John Wiley & Sons, New Jersey, 2008), p. 1038.
31. K. Lasanen, V. Korkala, E. Raisanen-Ruotsalainen and J. Kostamovaara, Design of a 1-V low power CMOS bandgap reference based on resistive subdivision, *Proc. 45th Midwest Symp. Circuits and Systems*, Tulsa, USA, August 2002, pp. 564–567.
32. K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker and S. Zizala, WiCkeD: Analog circuit synthesis incorporating mismatch, *Proc. Custom Integrated Circuits Conf.*, Orlando, USA, May 2000, pp. 511–514.