Published in IET Circuits, Devices & Systems
Received on 29th March 2010
Revised on 19th July 2010
doi: 10.1049/iet-cds.2010.0094

ISSN 1751-858X

Direct search approach to integrated circuit sizing for high parametric yield

G. Cijan[1] *T. Tuma*[2] *Á. Bűrmen*[3]

[1]*Regional development agency of Northern Primorska, Mednarodni prehod 6, SI-5290 Šempeter pri Gorici, Slovenia*
[2]*Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*
[3]*Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*
E-mail: gregor.cijan@rra-sp.si

Abstract: This study proposes a gradient-free approach to integrated circuit sizing that takes into account the statistical variations of device parameters and ranges of operating conditions. A novel gradient-free algorithm for solving the worst-case performance problem is proposed. The proposed algorithm produces corners that are used in the optimisation loop of the circuit sizing process. The set of corners is dynamically updated during circuit sizing. The number of corners is kept low by considering only corners that are sufficiently unique. The final result is a circuit exhibiting the specified parametric yield. The proposed approach was tested on several circuits and the results were verified with Monte–Carlo analysis and worst-case distances. All resulting circuits were obtained in up to 12 h on a single processor and exhibited the specified yield. The method can easily be parallelised to an extent that can bring the runtime of the method in the range of an hour or less.

1 Introduction

The importance of design automation in the area of integrated circuit (IC) design is constantly increasing. The design automation for analogue circuits is still not developed to the extent it has reached in the area of digital circuits. Therefore new methods are needed to remove the bottleneck in the design of analogue and mixed signal circuits. One of the essential tasks in this area is obtaining IC designs with high parametric yield (in the remainder of the paper also referred to as the yield), that is, designs that result in a high percentage of manufactured circuits satisfying all design requirements. Parametric yield does not take into account catastrophic faults.

This paper considers circuit sizing, that is, it assumes that the circuit topology is chosen in advance using designer's experience. By adjusting the design parameters of a preselected topology one attempts to achieve a satisfactory yield. The main cause for small yield are the random variations of the electrical properties of identically designed devices. The most problematic of these variations is the mismatch [1–3].

Several analogue design automation approaches were suggested in the past. Some of them dealt only with nominal design and neglected the issue of yield [4, 5]. More advanced techniques also automated the design of circuits with high yield [6–12]. Older approaches assumed that statistical variations are applied to design parameters only [6] which is not the case for ICs. Methods that use response surface modelling (e.g. [8, 10]) suffer from model accuracy problems. Often a linear model is used (e.g. [10]) which is inadequate for describing the behaviour of many important circuit properties (like mismatch-caused offset voltage). Performance models are almost always local by nature and cannot be used with global optimisation methods without frequently rebuilding the model.

The prerequisite for a successful yield optimisation is a reliable and accurate worst-case analysis, either in the form of worst-case performance (e.g. [9]) or worst-case distance (WCD) (e.g. [7, 10]). The main shortcoming of approaches based on [7, 11, 12] is that the worst-case analysis must be performed for every iteration of the yield optimisation process. Owing to the complexity of the worst-case analysis, this leaves little space for anything else but a local optimisation method for sizing the circuit (usually a trust region method with a linear model like in [11, 12]). Model-based approaches do not analyse real circuits in the course of optimisation and so the final optimisation result must be verified with a circuit analysis to make sure it is correct.

The worst case can be obtained using global optimisation methods (e.g. [9]). This approach is formally the most correct one. Unfortunately, global optimisation is computationally much more intensive than local optimisation. The results published in [9] show that the global approach produces a circuit with a relatively low yield (around 50%). This circuit is then used as a good starting point for a local yield optimisation approach like the one found in [10]. Owing to the method that is used simulated annealing (SA), [9] is a very time-consuming solution.

Gradient-based local algorithms are often used for solving the worst-case problem (e.g. [7, 11]) despite the fact that most simulators are incapable of calculating sensitivities. Such approaches require the numerical evaluation of performance

gradients from circuit performances which are often subject to a significant numerical error introduced by the simulator. Results on real-world circuits (e.g. [11–13]) have shown that a local worst-case evaluation method combined with sizing rules [14] and a local yield optimisation algorithm [15] produces good results despite the local nature of the underlying optimisation algorithms. This was explained with the fact that most circuit performances are only weakly non-linear in regions where sizing rules are satisfied.

An approach that requires no gradients or costly global methods for finding the worst-case performance is proposed in this paper. Response surface modelling is not used by the approach and so problems with model accuracy become irrelevant. The algorithm for finding the worst-case performance belongs to the family of direct search methods [16]. The proposed algorithm for finding the worst-case performance can be efficiently included in a yield optimisation process in a way similar to the one used in [9]. The resulting yield optimisation approach is tested on several IC cells and produces circuits with high yield (99%). In the process of yield optimisation a local or a global optimisation method can be used for sizing the circuit.

The remainder of the paper is structured as follows: Section 2 introduces the basic notions. Section 3 highlights the relation between WCDs [7] and yield whereas Section 4 establishes the relation between yield and worst-case performance. Sections 5 and 6 propose an approach to worst-case performance evaluation and apply it to yield optimisation. The proposed approach is evaluated on several test problems and verified in Section 7. Section 8 concludes the paper.

Notation: If $\boldsymbol{x}$ is a vector, then its components are denoted by x_i. $\angle(\boldsymbol{x}, \boldsymbol{y})$ is the angle between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the dot product of two vectors and $[\boldsymbol{a}, \boldsymbol{b}]$ defines a new vector composed of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

2 Circuit performance and yield

The performance of an analogue IC can be described as a set of performance measures (e.g. current consumption, gain, phase margin, delay, etc.). In real-world circuits, these values depend on three kinds of parameters [17]: design parameters ($\boldsymbol{x}_\mathrm{D}$), range parameters ($\boldsymbol{x}_\mathrm{R}$) and statistical parameters ($\boldsymbol{x}_\mathrm{S}$). A performance measure f_i is actually a function $f_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{R}, \boldsymbol{x}_\mathrm{S})$. Let n_D, n_R and n_S denote the number of design, range and statistical parameters, respectively.

The circuit's performance is satisfactory if all its m performance measures satisfy their respective design requirements. A design requirement is of the form $f_i \le F_i$ or $f_i \ge F_i$ where F_i is the ith target value. For the sake of simplicity only $f_i \ge F_i$ is considered. $f_i \le F_i$ can be transformed into $f_i \ge F_i$ by replacing f_i and F_i with $-f_i$ and $-F_i$.

Range parameter values come from a bounded set $\mathcal{R}$ which can be expressed as $\boldsymbol{x}_\mathrm{R} \in \mathcal{R}$. Some of the most common range parameters that describe the operating conditions of a circuit are temperature, supply voltage, bias current, etc. Usually $\mathcal{R}$ is a cross-product of n_R intervals where lower- and upper-interval bounds are specified by components of vectors $\boldsymbol{L}$ and $\boldsymbol{H}$, respectively. Every interval corresponds to one range parameter. Nominal range parameter values are denoted by $\boldsymbol{x}_\mathrm{R}^\mathrm{N}$.

Statistical parameters ($\boldsymbol{x}_\mathrm{S}$) model the variations of the manufacturing process. They are usually described by a joint probability distribution function (JPDF) and generally come from a set that is not bounded. A very common JPDF is the multivariate normal distribution

$$p(\boldsymbol{x}_\mathrm{S}) = \frac{1}{\left(\sqrt{2\pi}\right)^{n_\mathrm{S}}\sqrt{\det \boldsymbol{C}}} \exp\left(\frac{-\beta^2(\boldsymbol{x}_\mathrm{S})}{2}\right) \tag{1}$$

where $\beta^2(\boldsymbol{x}_\mathrm{S}) = (\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{S}^\mathrm{N})^\mathrm{T}\boldsymbol{C}^{-1}(\boldsymbol{x}_\mathrm{S} - \boldsymbol{x}_\mathrm{S}^\mathrm{N})$, $\boldsymbol{x}_\mathrm{S}^\mathrm{N}$ is the vector of mean statistical parameter values (also referred to as the nominal statistical parameters) and $\boldsymbol{C}$ denotes the covariance matrix. If the JPDF describing $\boldsymbol{x}_\mathrm{S}$ is continuous but not normal, it can be made normal by applying an appropriate transformation to $\boldsymbol{x}_\mathrm{S}$. For modern ICs this transformation depends on the design parameters $\boldsymbol{x}_\mathrm{D}$. One can always transform such JPDFs into a normal distribution with $\boldsymbol{x}_\mathrm{S}^\mathrm{N} = \boldsymbol{0}$ and $\boldsymbol{C} = \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix. Therefore without loss of generality the latter distribution is assumed throughout the remainder of this paper. A yield partition is defined as

$$Y_i(\boldsymbol{x}_\mathrm{D}) = \int_{\boldsymbol{x}_\mathrm{S} \in \mathbb{R}^{n_\mathrm{S}}} p(\boldsymbol{x}_\mathrm{S}) h_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{S})\, \mathrm{d}\boldsymbol{x}_\mathrm{S} \tag{2}$$

where $h_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{S})$ is the indicator function of the ith performance measure (i.e. it is 1 if $f_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{R}, \boldsymbol{x}_\mathrm{S}) \ge F_i$ for all $\boldsymbol{x}_\mathrm{R} \in \mathcal{R}$ and 0 otherwise). The total yield $Y(\boldsymbol{x}_\mathrm{D})$ can be defined in a similar way as (2) except that $h_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{S})$ is replaced by $\prod_{i=1}^{m} h_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{S})$. The job of a circuit designer is to choose the design parameters ($\boldsymbol{x}_\mathrm{D}$) in such manner that the circuit's total yield will be as large as possible.

3 Yield estimation and WCDs

The absolute value of the WCD $\beta_i^{\mathrm{WCD}}(\boldsymbol{x}_\mathrm{D})$ is defined as the shortest distance between the origin ($\boldsymbol{x}_\mathrm{S}^\mathrm{N}$) and the contour C defined as $\min_{\boldsymbol{x}_\mathrm{R} \in \mathcal{R}} f_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{R}, \boldsymbol{x}_\mathrm{S}) = F_i$ [7]. C represents the boundary of the acceptance region corresponding to Y_i (shaded in dark grey in Fig. 1) which is the part of the space of statistical parameters for which the indicator function $h_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{S})$ is equal to 1. The WCD is positive if $\min_{\boldsymbol{x}_\mathrm{R} \in \mathcal{R}} f_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{R}, \boldsymbol{x}_\mathrm{S}^\mathrm{N}) \ge F_i$ holds and negative otherwise.

If the acceptance region boundary is approximated with a halfspace tangent to C at the point closest to the origin, the yield partition suffers an error (acceptance region approximation error) denoted by the hashed region in

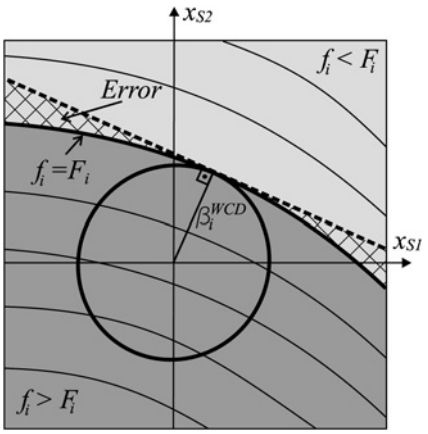

Fig. 1 *WCD problem illustrated for two statistical parameters*

Contours of $\min_{\boldsymbol{x}_\mathrm{R} \in \mathcal{R}} f_i(\boldsymbol{x}_\mathrm{D}, \boldsymbol{x}_\mathrm{R}, \boldsymbol{x}_\mathrm{S})$ for a fixed set of design parameters $\boldsymbol{x}_\mathrm{D}$ are depicted by solid curves. The acceptance region of f_i is shaded in dark grey

doi: 10.1049/iet-cds.2010.0094

Fig. 1. This error proves to be reasonably small in IC design [17]. A yield partition can be approximated as [17]

$$Y_i(\boldsymbol{x}_{\mathrm{D}}) \simeq \Theta(\beta_i^{\mathrm{WCD}}(\boldsymbol{x}_{\mathrm{D}})) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\beta_i^{\mathrm{WCD}}(\boldsymbol{x}_{\mathrm{D}})}{\sqrt{2}}\right)\right) \quad (3)$$

As all yield partitions Y_i approach 1 the total yield Y also approaches 1. The yield can therefore be maximised by maximising the smallest yield partition which is equivalent to maximising the smallest WCD. Maximisation of WCDs pushes the boundaries of the acceptance region away from the origin. Owing to the nature of the underlying optimisation problem, the computation of WCD requires a gradient-based modified SQP (sequential quadratic programming) algorithm [7, 17, 18].

4 Worst-case performance and its relation to yield

Instead of maximising the smallest WCD, let us require that WCDs be greater than some $\beta^{\mathrm{WPM}} > 0$. This is equivalent to requiring that the worst performance inside the sphere $\beta \leq \beta^{\mathrm{WPM}}$ satisfies the design requirements. To optimise the yield by improving worst performance measures (WPMs) an algorithm for calculating the worst-case performance is needed. Finding the WPM is a much simpler problem than the computation of the WCD. The WPM problem can be stated as

$$f_i^{\mathrm{WPM}} = \min_{\boldsymbol{x}_{\mathrm{R}} \in \mathcal{R}, \beta \leq \beta^{\mathrm{WPM}}} f_i(\boldsymbol{x}_{\mathrm{D}}, \boldsymbol{x}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{S}}) \quad (4)$$

Ensuring that the worst-case performance inside the sphere $\beta \leq \beta^{\mathrm{WPM}}$ satisfies the design requirements is equivalent to ensuring that the WCD is greater than β^{WPM}. This implies (within acceptance region approximation error) that the yield partition satisfies

$$Y_i(\boldsymbol{x}_{\mathrm{D}}) \geq \Theta(\beta^{\mathrm{WPM}}) \quad (5)$$

Two examples of the WPM problem are shown in Fig. 2 where the two WPM solutions f_i^{W} and $f_i^{\mathrm{W}'}$ are obtained using β^{WPM} and $\beta^{\mathrm{WPM}'}$. By setting $\beta^{\mathrm{WPM}} = 3$ the yield partitions are required to satisfy $Y_i \geq 0.9987$.

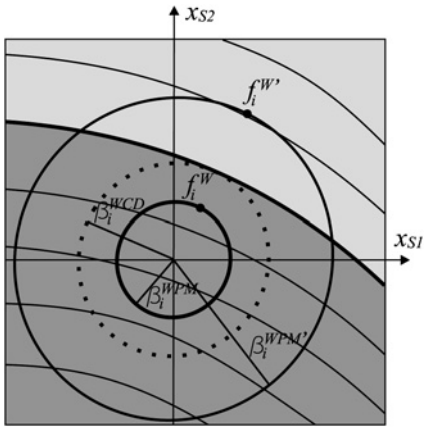

Fig. 2 *Two WPM problems, one for $\beta_i^{WPM} < \beta_i^{WCD}$ and one for $\beta_i^{WPM'} > \beta_i^{WCD}$*

Solution (f_i^{W} and $f_i^{\mathrm{W}'}$, respectively) is denoted by a dot. The two solutions satisfy $f_i^{\mathrm{W}} > F_i > f_i^{\mathrm{W}'}$. See Fig. 1 for the explanation of contours and shading

5 Solving the worst-case performance problem

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 an unconstrained direct search method to the nature of these constraints. The proposed algorithm draws its inspiration from the Hooke–Jeeves method [19]. The method uses two types of steps. Trial steps search for a lower value of the function f in the neighbourhood 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. The size of this step is determined by the α parameter.

The search steps in the space of range parameters are identical to Hooke–Jeeves steps, whereas the search steps in the space of statistical parameters are rotations and radial steps. This makes sure that the steps conform to the boundary of the search space. The step acceptance criterion in the space of statistical parameters requires sufficient descent (i.e. a sufficiently large decrease of the function that is subject to minimisation) [20].

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ denote the space of statistical and range parameters, respectively. The WPM problem for finding the set of statistical and range parameters at which worst-case performance occurs can be stated mathematically as

$$\min_{\boldsymbol{x} \in \mathbb{R}^{n_{\mathrm{S}}+n_{\mathrm{R}}}} f(\boldsymbol{x}) \text{ subject to} \quad (6)$$

$$\|\boldsymbol{x}_{\mathrm{S}}\| \leq \beta, \quad \beta > 0 \quad (7)$$

$$L_i \leq x_{\mathrm{R}i} \leq H_i, \quad i = 1, \ldots, n_{\mathrm{R}} \quad (8)$$

For $\boldsymbol{x} \in \mathcal{S}_1 \times \mathcal{S}_2$ vectors $\boldsymbol{x}_{\mathrm{S}}$ and $\boldsymbol{x}_{\mathrm{R}}$ represent the components of $\boldsymbol{x}$ corresponding to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Vector $\boldsymbol{x}$ can be written as $[\boldsymbol{x}_{\mathrm{S}}, \boldsymbol{x}_{\mathrm{R}}]$ where $\boldsymbol{x}_{\mathrm{S}}$ and $\boldsymbol{x}_{\mathrm{R}}$ are vectors with n_{S} and n_{R} components, respectively.

Basis vectors for $\mathcal{S}_1$ and $\mathcal{S}_2$ are denoted by $\boldsymbol{e}^i$ and $\boldsymbol{b}^i$, respectively. All basis vectors are mutually orthogonal with unit length ($\|\boldsymbol{e}^i\| = 1$, $\|\boldsymbol{b}^i\| = 1$). Every basis vector corresponds to one statistical or one range parameter. Fig. 3 depicts rotations (φ) and radial steps (R) in $\mathcal{S}_1$ and coordinate steps (Φ) in $\mathcal{S}_2$. Rotations enable the algorithm to move on the surface of a hypersphere defined by $\|\boldsymbol{x}_{\mathrm{S}}\| \leq \beta$

$$\boldsymbol{x}'_{\mathrm{S}} = \mathrm{rotate}(\boldsymbol{x}_{\mathrm{S}}, \boldsymbol{e}, \varphi) = \begin{cases} \boldsymbol{x}_{\mathrm{S}} \cos\varphi + \dfrac{\boldsymbol{e}'}{\|\boldsymbol{e}'\|}\|\boldsymbol{x}_{\mathrm{S}}\| \sin\varphi \\ \quad \boldsymbol{x}_{\mathrm{S}} \text{ and } \boldsymbol{e} \text{ linearly independent} \\ \boldsymbol{x}_{\mathrm{S}}; \text{ otherwise} \end{cases} \quad (9)$$

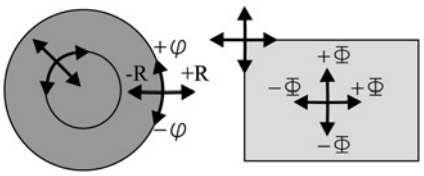

Fig. 3 *Rotations and radial steps at two points in subspace $\mathcal{S}_1$ (left), coordinate steps at two points in subspace $\mathcal{S}_2$ (right)*

where $\boldsymbol{e}'$ is obtained as

$$\boldsymbol{e}' = \boldsymbol{e} - \frac{\boldsymbol{e} \cdot \boldsymbol{x}_S}{\|\boldsymbol{x}_S\|^2} \boldsymbol{x}_S \tag{10}$$

Radial steps move between hyperspheres in $\mathcal{S}_1$ with different radii

$$\begin{aligned} \boldsymbol{x}_S^* &= \text{radial}(\boldsymbol{x}_S, R, \beta_{\min}) \\ &= \begin{cases} \boldsymbol{x}_S + \dfrac{\boldsymbol{x}_S}{\|\boldsymbol{x}_S\|} R; & \|\boldsymbol{x}_S + \dfrac{\boldsymbol{x}_S}{\|\boldsymbol{x}_S\|} R\| \geq \beta_{\min} \\ -\dfrac{\boldsymbol{x}_S}{\|\boldsymbol{x}_S\|} \beta_{\min}; & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

The introduction of $\beta_{\min}$ prevents $\|\boldsymbol{x}_S\|$ from becoming 0. Radial steps that result in a point inside the hypersphere with radius $\beta_{\min}$ are modified so that the resulting point lies on the surface of this hypersphere.

The following definitions simplify the description of the algorithm. The step size is defined as

$$\Delta_i = \begin{cases} \varphi; & 1 \leq i \leq n_S \\ R; & i = n_S + 1 \\ \Phi(H_{i-(n_S+1)} - L_{i-(n_S+1)}); & \text{otherwise} \end{cases} \tag{12}$$

where φ, R and Φ denote the size of the rotational, radial and coordinate steps, respectively. i denotes the index assigned to an individual step.

A rotational step is successful if it produces sufficient descent (γ). Successful radial steps and successful coordinate steps must produce simple descent (i.e. $f(\boldsymbol{x}^{\text{temp}}) < f^{\text{T}}$). d_i denotes the amount of descent that is sufficient for ith step to be considered successful.

$$d_i = \begin{cases} \gamma; & 1 \leq i \leq n_S \\ 0; & n_S + 1 \leq i \leq n_S + n_R + 1 \end{cases} \tag{13}$$

The actual steps ($\boldsymbol{P}(\boldsymbol{x}, \Delta, i)$) are defined as

$$\begin{aligned} &\boldsymbol{P}(\boldsymbol{x}, \Delta, i) \\ &= \begin{cases} [\text{rotate}(\boldsymbol{x}_S, \boldsymbol{e}^i, \Delta), \boldsymbol{x}_R]; & 1 \leq i \leq n_S \\ [\text{radial}(\boldsymbol{x}_S, \Delta, \beta_{\min}), \boldsymbol{x}_R]; & i = n_S + 1 \\ [\boldsymbol{x}_S, \boldsymbol{x}_R + \Delta \boldsymbol{b}^{i-(n_S+1)}]; & n_S + 1 < i \leq n_S + n_R + 1 \end{cases} \end{aligned} \tag{14}$$

Constraints (7) and (8) can be violated by certain steps. For this purpose, a vector-valued function $\boldsymbol{\Omega}(\boldsymbol{x})$ is defined which moves a point in such a manner that the constraints are satisfied again.

$$\boldsymbol{\Omega}(\boldsymbol{x}) = \left[\boldsymbol{\Omega}_S(\boldsymbol{x}_S), \sum_{i=1}^{n_R} (\Omega_R(x_{Ri}, L_i, H_i) \boldsymbol{b}^i) \right] \tag{15}$$

$$\boldsymbol{\Omega}_S(\boldsymbol{x}_S) = \begin{cases} \boldsymbol{x}_S; & \|\boldsymbol{x}_S\| \leq \beta \\ \dfrac{\boldsymbol{x}_S}{\|\boldsymbol{x}_S\|} \beta; & \|\boldsymbol{x}_S\| > \beta \end{cases} \tag{16}$$

Choose $\mathbf{x}^0$ satisfying (7) and (8).
Choose $\varphi > 0$, $R > 0$, $\Phi > 0$, $\gamma > 0$, $0 < \beta_{min} < \beta$.
Choose $r > 1$, $\delta > 1$ 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
    $sp := 1$;
    $\mathbf{x}_S^{SP} := \text{rotate}\left(\mathbf{x}_S^B, \mathbf{x}_S^T, \alpha \angle(\mathbf{x}_S^T, \mathbf{x}_S^B)\right)$
    $\mathbf{x}_S^{SP} := \text{radial}\left(\mathbf{x}_S^{SP}, \alpha(\|\mathbf{x}_S^T\| - \|\mathbf{x}_S^B\|), \beta_{min}\right)$;
    $\mathbf{x}_R^{SP} := \mathbf{x}_R^B + \alpha(\mathbf{x}_R^T - \mathbf{x}_R^B)$;
    $\mathbf{x}^B := \mathbf{x}^T$; $f^B := f^T$;
    $\mathbf{x}^T := \Omega(\mathbf{x}^{SP})$;
  else
    // speculative step failed
    $sp := 0$;
    $\mathbf{x}^T := \mathbf{x}^B$; $f^T := f^B$;
  end
  for $(i := 1, n_S + 2, 2, 3, ..., n_S, n_S + 1,$
      $n_S + 3, n_S + 4, ..., n_S + n_R + 1)$ **do**
    // trial step (+)
    $\mathbf{x}^{temp} := \Omega(\mathbf{P}(\mathbf{x}^T, +\Delta_i, i))$;
    $flag := 1$;
    if $f(\mathbf{x}^{temp}) < f^T - d_i$ **then**
      // trial step (+) accepted
      $\mathbf{x}^T := \mathbf{x}^{temp}$; $f^T := f(\mathbf{x}^{temp})$;
      $flag := 0$;
    end
    if $flag = 1$ **or** $sp = 1$ **then**
      // trial step (-)
      $\mathbf{x}^{temp} := \Omega(\mathbf{P}(\mathbf{x}^T, -\Delta_i, i))$;
      if $f(\mathbf{x}^{temp}) < f^T - d_i$ **then**
        // trial step (-) accepted
        $\mathbf{x}^T := \mathbf{x}^{temp}$; $f^T := f(\mathbf{x}^{temp})$;
      end
    end
    if sp = 1 **and** $f^T \geq f^B$ **and** $i = n_S + 2$ **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
end

Fig. 4 *WPM algorithm*

doi: 10.1049/iet-cds.2010.0094

$$\Omega_{\mathrm{R}}(x, L, H) = \begin{cases} x; & L \le x \le H \\ L; & x < L \\ H; & x > H \end{cases} \quad (17)$$

Let $\boldsymbol{x}^0$ denote the initial point. The proposed algorithm for solving the WPM problem can now be represented by Fig. 4.

The upper index of $\boldsymbol{x}$ represents different points that are calculated by the algorithm. $\boldsymbol{x}^{\mathrm{T}}$ and $\boldsymbol{x}^{\mathrm{B}}$ represent the trial point and the best point while the corresponding values of f are denoted by f^{T} and f^{B}, respectively. The point produced by the speculative step is denoted by $\boldsymbol{x}^{\mathrm{SP}}$. The lower index of vector $\boldsymbol{x}$ denotes the sub-vector of statistical parameters ($\boldsymbol{x}_{\mathrm{S}}$) and range parameters ($\boldsymbol{x}_{\mathrm{R}}$). If there is no lower index, then vector $\boldsymbol{x}$ represents a point from $\mathcal{S}_1 \times \mathcal{S}_2$ and can be written as $[\boldsymbol{x}_{\mathrm{S}}, \boldsymbol{x}_{\mathrm{R}}]$. α specifies the scaling factor for the length of the speculative step. r and δ are used for changing the step size and the amount of sufficient descent.

If $n_{\mathrm{S}} = 0$ (or $n_{\mathrm{R}} = 0$), the sequence of indices traversed by the for statement is $2, 3, \ldots, n_{\mathrm{R}} + 1$ (or $1, 2, \ldots, n_{\mathrm{S}}, n_{\mathrm{S}} + 1$). For $n_{\mathrm{R}} = 0$ the condition in the 'if' statement that triggers the 'break' changes from $i = n_{\mathrm{S}} + 2$ to $i = 1$.

The implementation used the following initial parameter values: $\varphi = \pi/4$, $R = \beta/2$, $\Phi = 1/8$, $\beta_{\min} = \beta/3$, $r = 6$, $\alpha = 2$ and $\delta = 2$. The algorithm was stopped when $\Phi < 1/72$. This combination of parameters was obtained with numerical trials. The proposed values may not be the ultimate best choice but nevertheless resulted in satisfactory performance of the proposed algorithm.

For the initial range parameter values ($\boldsymbol{x}_{\mathrm{R}}^0$) the performance was measured at L_i and H_i for every range parameter while other components of $\boldsymbol{x}_{\mathrm{R}}$ remained nominal and $\boldsymbol{x}_{\mathrm{S}}$ was set to $\boldsymbol{x}_{\mathrm{S}}^{\mathrm{N}}$. Based on the results, extreme range parameter values were combined into vector $\boldsymbol{x}_{\mathrm{R}}^{\mathrm{A}}$ where the worst performance was expected. Finally, the WPM algorithm was run with $n_{\mathrm{S}} = 0$ and $\boldsymbol{x}^0 = \boldsymbol{x}_{\mathrm{R}}^{\mathrm{A}}$ resulting in $\boldsymbol{x}_{\mathrm{R}}^0$.

For obtaining $\boldsymbol{x}_{\mathrm{S}}^0$ and γ the $\boldsymbol{x}_{\mathrm{R}}$ was set to $\boldsymbol{x}_{\mathrm{R}}^0$ and the performance was evaluated at $x_{Si} = \pm\beta$ in a similar manner as it was done before with range parameters. The results (f^{+i} and f^{-i}) are used for calculating the gradient $\nabla_{\mathrm{S}} f$ at $\boldsymbol{x}_{\mathrm{S}}^{\mathrm{N}}$ via central difference approximation. Let $f^{\max}$ and $f^{\min}$ represent the largest and the smallest value of f across all $f^{\pm i}$. γ and $\boldsymbol{x}_{\mathrm{S}}^0$ are then obtained as $(f^{\max} - f^{\min})/10$ and $-\beta\nabla_{\mathrm{S}} f / \|\nabla_{\mathrm{S}} f\|$, respectively.

Despite the fact that the algorithm is local, it is expected to be adequate because the modified SQP approach to the WCD problem in [7, 17] is also local but still produces good results on real-world IC designs.

6 Direct search yield optimisation

A corner c is defined as a two-tuple $(\boldsymbol{x}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{S}})$. Let c^N denote the nominal corner $(\boldsymbol{x}_{\mathrm{R}}^{\mathrm{N}}, \boldsymbol{x}_{\mathrm{S}}^{\mathrm{N}})$. The set of corners corresponding to f_i is denoted as $\mathcal{C}_i$. Let $\boldsymbol{x}_{\mathrm{D}}^0$ denote the initial design parameters. The yield optimisation algorithm can now be represented as Fig. 5.

The optimisation (step A) starts with initial point $\boldsymbol{x}_{\mathrm{D}}$. The cost function for optimisation is constructed in such a manner that if measure f_i fails to satisfy $f_i \ge F_i$ in any of the corners from $\mathcal{C}_i$, a penalty proportional to the violation is added to the cost function [21]. The optimisation stops when the cost function reaches 0 (i.e. when all design requirements are fulfilled across all corresponding corners).

After the optimisation is finished, the obtained design ($\boldsymbol{x}_{\mathrm{D}}$) is checked for new corners (i.e. the worst-case performance

```
C_i := {c^N},  i = 1, 2, ..., m;
x_D := x_D^0;
stop:=False;
while not stop do
  stop:=True;
  x_D := Optimise(x_D, C_1, C_2, ..., C_m); // step A
  for i := 1, ..., m do
    (c_i^new, f_i^W) := WPM(x_D, i);
    if f_i^W < F_i then
      if ∃c ∈ C_i such that c ≈ c_i^new then
        C_i := C_i − {c};
      end
      C_i := C_i ∪ {c_i^new};
      stop:=False;
    end
  end
end
```

Fig. 5 *Yield optimisation algorithm*

f_i^{W} at $\boldsymbol{x}_{\mathrm{D}}$ is calculated for all performance measures using the proposed WPM algorithm). This results in m new corners c_i^{new}. If f_i^{W} fails to satisfy $f_i^{\mathrm{W}} \ge F_i$ the corresponding new corner c_i^{new} is added to the set of corners $\mathcal{C}_i$. If there exists a $c \in \mathcal{C}_i$ such that $c \simeq c_i^{\mathrm{new}}$ then c is removed from $\mathcal{C}_i$ before c_i^{new} is added.

Two corners $(c_1 = (\boldsymbol{x}_{\mathrm{R}}^1, \boldsymbol{x}_{\mathrm{S}}^1)$ and $c_2 = (\boldsymbol{x}_{\mathrm{R}}^2, \boldsymbol{x}_{\mathrm{S}}^2))$ are approximately equal ($c_1 \simeq c_2$) if all of the following holds

$$\angle(\boldsymbol{x}_{\mathrm{S}}^1, \boldsymbol{x}_{\mathrm{S}}^2) \le 15^\circ \quad (18)$$

$$|\|\boldsymbol{x}_{\mathrm{S}}^1\| - \|\boldsymbol{x}_{\mathrm{S}}^2\|| \le 0.25 \max(\|\boldsymbol{x}_{\mathrm{S}}^1\|, \|\boldsymbol{x}_{\mathrm{S}}^2\|) \quad (19)$$

$$|x_{\mathrm{R}i}^1 - x_{\mathrm{R}i}^2| \le 0.1 \max(|x_{\mathrm{R}i}^1|, |x_{\mathrm{R}i}^2|) \quad (20)$$

The last inequality must hold for all $i = 1, \ldots, n_{\mathrm{R}}$. The algorithm gradually builds the set of corners corresponding to individual performance measures. It is finished when the worst-case performance satisfies all design requirements. At that point, final yield partitions satisfy (5).

The WPM algorithm requires no derivatives for its operation. If a direct search optimisation algorithm is used in step A the yield optimisation algorithm does not require the evaluation of derivatives.

7 Examples

The proposed WPM-based yield optimisation approach was demonstrated on four different IC cells: an operational amplifier (OPAMP) [22], a bandgap reference circuit (BGR) [23], a beta multiplier reference (BMR) [22] and a comparator (COMPAR) [22]. The target yield was 99.87% ($\beta^{\mathrm{WPM}} = 3$).

Two range parameters were common to all four circuits: temperature (-20–80°C) and supply voltage (1.6–2.0 V). The OPAMP had an additional range parameter (bias current) ranging from 80 to 120 μA.

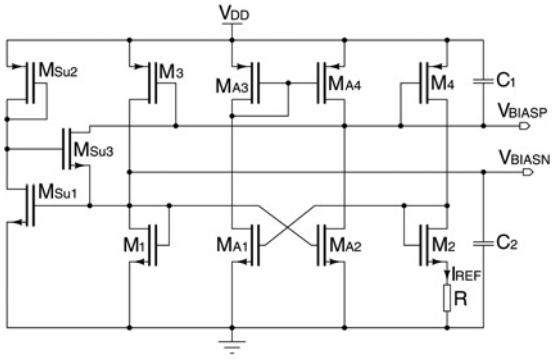

Fig. 6 *Beta-multiplier reference*

Four parameters of every MOS transistor were affected by mismatch: threshold voltage, channel width reduction, channel length reduction and oxide thickness. The vector of these parameters for all transistors in a circuit is denoted by $\boldsymbol{p}$ and can be expressed as

$$\boldsymbol{p} = \boldsymbol{\mu}_{\mathrm{p}} + \boldsymbol{A}_{\mathrm{p}}(W, L)\boldsymbol{x}_{\mathrm{S}} \tag{21}$$

where $\boldsymbol{\mu}_{\mathrm{p}}$ is a vector of mean parameter values while matrix $\boldsymbol{A}_{\mathrm{p}}(W, L)$ represents the mismatch model. Note that $\boldsymbol{\mu}_{\mathrm{p}}$ is constant while $\boldsymbol{A}_{\mathrm{p}}$ depends on the transistor channel widths and lengths. The details of the mismatch model are proprietary and cannot be published. $\boldsymbol{x}_{\mathrm{S}}$ denotes the vector of statistical parameters which in turn are uncorrelated normal random variables with zero mean and standard deviation 1.

The WPM algorithm and the yield optimisation were implemented using Python [24] and NumPy [25]. All circuits were simulated with the SPICE OPUS simulator [26] and a 0.18 μm technology model. The nominal values of all statistical parameters were 0.

The first tested circuit is the BMR circuit in Fig. 6. This is a current source that pushes the current (I_{REF}) through resistor R. This current is fairly stable with respect to temperature and supply voltage variations. One can mirror the reference current using gate voltages V_{BIASP} and V_{BIASN}. This circuit has nine design parameters ($n_{\mathrm{D}} = 9$): the resistance of R and the length and the width of all MOS transistors excluding the start-up circuit (M_{SU1}, M_{SU2} and M_{SU3}). Several transistors must have matching channel dimensions (i.e. M_1 and M_2, M_3 and M_4, M_{A1} and M_{A2} and M_{A3} and M_{A4}) and so MOS transistors contribute only eight design parameters instead of 16.

During the optimisation, two range parameters ($n_{\mathrm{R}} = 2$) and 32 statistical parameters ($n_{\mathrm{S}} = 32$) were considered. The range parameters were the temperature (-20–80°C) and the supply voltage (1.6–2.0 V). The nominal values of range parameters were 27°C and 1.8 V, respectively. The 32 statistical parameters were contributed by transistors $M_1 - M_4$ and $M_{A1} - M_{A4}$.

In this circuit three circuit performances were considered. The first two were $\Delta I_{\mathrm{REF}}/\Delta V_{\mathrm{DD}}$ and $\Delta I_{\mathrm{REF}}/\Delta T$ that reflected the maximal output current variation with respect to supply voltage and temperature variations, respectively. The third measure was the value of the reference current that flows through resistance R. The goals for these measures were $\Delta I_{\mathrm{REF}}/\Delta V_{\mathrm{DD}} < 10\,\mathrm{nA/V}$, $\Delta I_{\mathrm{REF}}/\Delta T < 45\,\mathrm{nA/^{\circ}C}$ and $15\,\mu\mathrm{A} < I_{\mathrm{REF}} < 30\,\mu\mathrm{A}$. To speed up the optimisation, some additional performance

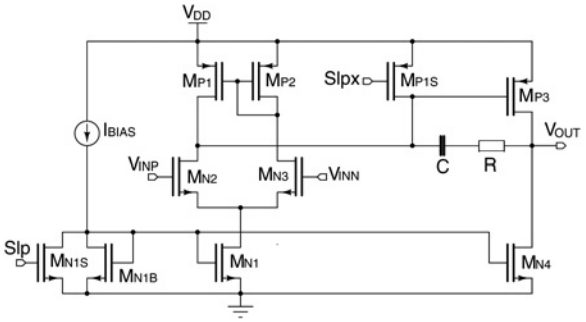

Fig. 7 *Operational amplifier*

measures were considered for transistors $M_1 - M_4$ and $M_{A1} - M_{A4}$. The goals for these measures were $V_{\mathrm{ds}} - V_{\mathrm{th}} > 0$ and $V_{\mathrm{ds}} - V_{\mathrm{dsat}} > 0$. In [14], such measures are referred to as sizing rules. The sizing rules were not subject to WPM analysis and yield optimisation.

The remaining three examples are set up in a way similar to the BMR. The OPAMP is depicted in Fig. 7. It has 32 statistical (four for every transistor with the exception of M_{N1S} and M_{P1S}) and three range parameters. The bandgap reference circuit on Fig. 8 is designed for low-power CMOS and is based on resistive subdivision. It has 40 statistical parameters. Transistors M_4, M_5 and M_6 comprise the start up circuit and contribute no statistical parameters. In the COMPAR circuit (Fig. 9), variations of statistical parameters cause a significant increase of hysteresis. The

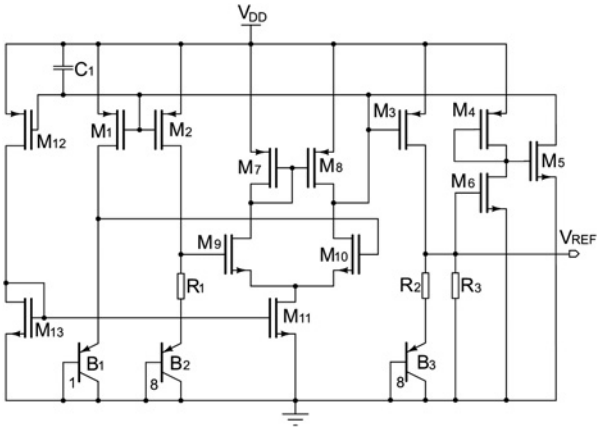

Fig. 8 *Bandgap reference*

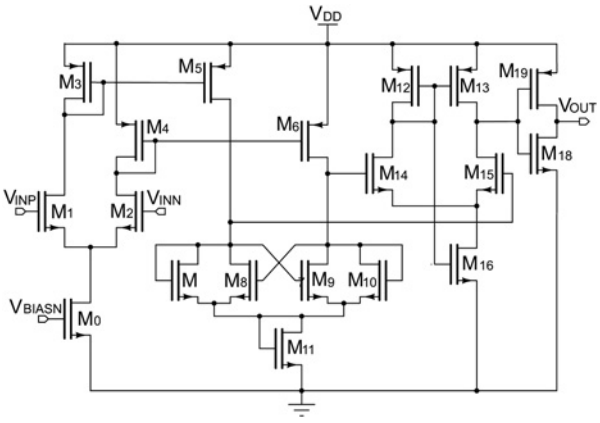

Fig. 9 *Comparator*

doi: 10.1049/iet-cds.2010.0094

BMR circuit (see Fig. 6) was used as a source of reference current for the COMPAR. The complete COMPAR has 108 statistical parameters (32 because of the BMR and 76 because of the COMPAR).

The initial set of design parameters $\boldsymbol{x}_{\mathrm{D}}^{0}$ for every circuit was obtained with the parallel simulated annealing with differential evolution (PSADE) global optimisation method by considering only the nominal corner. PSADE is a parallel version of the DESA global optimisation algorithm [27], which performs well on various mathematical and IC optimisation problems. The initial point for all four circuits was obtained in a few minutes on a parallel computer system of five 3 GHz dual-core computers. The initial point obtained with such a global method is a good starting point for the direct search yield optimisation algorithm presented in the previous section.

Modified Box simplex algorithm [28] was used for the optimisation in step A. We used this local optimisation method because experiments have shown that with a good starting point (one that satisfies the design requirements in the nominal corner) it produces results that are as good as those obtained with the PSADE algorithm, except that the Box simplex algorithm is faster. The results of yield optimisation are listed in Tables 1–4. The first three columns of Table 1 list the name, the type and the goal (F_i) of the performance measures, respectively. The fourth, the fifth and the sixth column list the initial performances in the nominal corner ($f_{\mathrm{INI}}^{\mathrm{N}}$), the initial worst-case performances ($f_{\mathrm{INI}}^{\mathrm{WPM}}$) and the worst-case performances obtained after yield optimisation ($f_{\mathrm{OPT}}^{\mathrm{WPM}}$). From the table it is clear that all worst-case performance measures of all four circuits have satisfied the respective goals after yield optimisation. Therefore all yield partitions of all circuits should be above 99.87%.

Table 1 Results of the yield optimisation algorithm

Performance name	Type	F_i	$f_{\mathrm{INI}}^{\mathrm{N}}$	$f_{\mathrm{INI}}^{\mathrm{WPM}}$	$f_{\mathrm{OPT}}^{\mathrm{WPM}}$
BMR					
$\frac{\Delta I_{\mathrm{REF}}}{\Delta T}$, nA/°C	<	45	45	46	45
$\frac{\Delta I_{\mathrm{REF}}}{\Delta V_{\mathrm{DD}}}$, nA/V	<	10	0.2	0.7	9.8
OPAMP					
slew rate rise, V/μs	>	4.0	6.9	3.9	4.2
slew rate fall, V/μs	>	4.0	12.3	4.2	17
settling time rise, μs	<	1.0	0.3	0.4	0.4
settling time fall, μs	<	1.0	0.1	0.3	0.07
rise time, ns	<	250	137	207	190
fall time, ns	<	250	98	215	49
overshoot, %	<	10	0.18	2.7	5.4
undershoot, %	<	10	0.01	0.02	0.01
swing 50% gain, V	>	1.0	1.2	0.95	1.1
phase margin, deg	>	55	57	54	68
unity gain BW, MHz	>	8.0	20	15	14
gain at 0 Hz, dB	>	60	82	74	79
out offset (upper), mV	<	4	0	5.5	3.7
out offset (lower), mV	>	−4	0	−5.5	3.7
BGR					
$\frac{\Delta V_{\mathrm{REF}}}{\Delta T}$, μV/°C	<	50	18	70	37
$\frac{\Delta V_{\mathrm{REF}}}{\Delta V_{\mathrm{DD}}}$, mV/V	<	5	4.4	17	4.1
COMPAR					
output rise time, ns	<	2	1.1	1.4	0.7
output fall time, ns	<	2	0.7	1.1	1.2
output rise delay, ns	<	15	6.8	19	13
output fall delay, ns	<	15	8.0	15	9
hysteresis, mV	<	4.0	3.2	37	3.4

Table 2 WCD and yield partitions

Performance name	WCD	Y_i, %
BMR		
$\frac{\Delta I_{\mathrm{REF}}}{\Delta T}$, nA/°C	3.01	99.85
$\frac{\Delta I_{\mathrm{REF}}}{\Delta V_{\mathrm{DD}}}$, nA/V	3.03	99.87
OPAMP		
slew rate rise, V/μs	5.03	100
out offset (upper), mV	3.15	99.91
out offset (lower), mV	3.15	99.94
BGR		
$\frac{\Delta V_{\mathrm{REF}}}{\Delta T}$, μV/°C	4.49	100
$\frac{\Delta V_{\mathrm{REF}}}{\Delta V_{\mathrm{DD}}}$, mV/V	3.75	99.99
COMPAR		
hysteresis, mV	3.42	99.94

Table 3 Number of optimisation parameters

Circuit	n_{S}	n_{R}	n_{D}
BMR	32	2	9
OPAMP	32	3	10
BGR	40	2	13
COMPAR	108	2	31

Table 4 Comparison of the proposed algorithm and the algorithm from [9]

Circuit	Proposed algorithm			SA algorithm		
	Y, %	n_{sim}	T, h	Y, %	n_{sim}	T, h
BMR	99.73	25 735	1.0	96.24	34 917	1.9
OPAMP	99.85	31 682	1.2	84.63	232 054	12.0
BGR	99.99	7867	1.6	95.57	61 113	13.5
COMPAR	99.94	72 772	8.4	87.12	79 209	11.4

The final results were verified by Monte–Carlo (MC) analysis of 10 000 random samples in the space of statistical parameters. For every sample, a WPM analysis with $n_{\mathrm{S}} = 0$ was performed resulting in worst-case performances over range parameters for the given set of statistical parameters. The WPM analysis results were used to obtain the actual yield partitions (Y_i) (last column of Table 2). The WCDs for the final results were also computed and are listed in the second column of Table 2. The table lists only those circuit performance measures for which the WCD is below six. It can be seen that no WCD was below three which once again confirms the final yield

partitions. The yield partition corresponding to beta-multiplier temperature dependence (99.85%) was slightly lower than the target yield (99.87%). This can be attributed to the fact that the circuit optimisation algorithm was stopped as soon as all worst-case performances satisfied their respective goals. Owing to the numerical errors caused by the simulator, some of them were slightly below the goal resulting in a slightly lower yield partition than expected.

Table 3 lists the number of optimisation variables considered in the yield optimisation process and in the proposed WPM algorithm. The number of optimisation variables in the yield optimisation process is equal to the number of design parameters (n_D). On the other hand, the number of optimisation variables in the proposed WPM algorithm is equal to the sum of the number of range and statistical parameters ($n_R + n_S$).

The yield optimisation was not parallelised and was executed on a single 3 GHz Intel processor. One analysis in one corner was considered as one circuit simulation. For every set of design parameters examined by the optimiser, a performance measure contributed as many simulations as there were corners in its corresponding $\mathcal{C}_i$. With this in mind, the cost evaluation can be parallelised to great extent. For example, in a problem with five analyses where every analysis must be performed across three corners, a total of 15 simulations is needed. If this problem is run on a parallel system of 15 or more processors, then the time needed for performing 15 simulations is equal to the time of the longest simulation and some additional time arising from communication overhead. In the same way, the WPM evaluations of individual performances can also be run in parallel for all measures at once. This has the potential of speeding up the algorithm proportionally to the number of performance measures.

The comparison of the proposed algorithm with the approach presented in [9] is shown in Table 4. The main difference between the two approaches is in the calculations of the WPM. In our approach, the algorithm presented in Section 5 is used, while [9] uses SA. The starting point and the initial circuit optimisation in the nominal corner were identical for both cases.

The second and fifth columns in Table 4 list the final yield obtained with the proposed method and the method presented in [9]. The final yields were obtained in the same manner as the yield partitions in Table 2 (10 000 sample MC analysis). The final yield of the circuits obtained with the proposed approach is close to the desired yield (99.87%). On the other hand the circuits obtained with the approach from [9] (with the exception of the BGR and BMR) exhibit a significantly lower yield. For these circuits a further yield optimisation step (e.g. using the LPP method [10], as noted in [9]) is required to reach the desired yield.

We tried to tighten the stopping criterion of SA in the hope of obtaining final yields that would be closer to the desired yield. The result was an improvement of the final yields (which unfortunately were still lower than the desired yield), but the algorithm runtimes increased by an order of magnitude.

The proposed method is also faster than [9], both in terms of the number of simulations (n_{sim}) as in terms of runtime (T). The average runtime of the algorithm per simulation is longer for SA than for the proposed algorithm. This can be attributed to the fact that because of its random nature the SA used in [9] generates a large number of circuits with extremely bad performance. Such circuits usually result in long transient analysis runtimes.

8 Conclusion

An approach to circuit sizing that takes yield into account was proposed. The approach is direct search based and does not require the evaluation of derivatives (sensitivities). It utilises a novel direct search approach for worst-case performance evaluation and a standard direct search optimisation algorithm (Box simplex algorithm) for circuit sizing. Algorithm runs on four examples of IC cell design have shown that the desired yield (i.e. 99.87%) can be achieved in hours on a single processor. The final yield was verified by simulating 10 000 MC samples and evaluating the final design's WCDs. Both tests confirmed the results. The algorithm can easily be parallelised. The expected maximal speed up is proportional to the number of performance measures subject to optimisation.

9 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 optimisation methods in telecommunications.

10 References

1 Michael, C., Ismail, M.: 'Statistical modeling of device mismatch for analog MOS integrated circuits', *IEEE J. Solid-State Circuits*, 1992, **27**, (2), pp. 154–166
2 Pelgrom, M.J.M., Duinmaijer, A.C.J., Welbers, A.P.G.: 'Matching properties of MOS transistors', *IEEE J. Solid-State Circuits*, 1989, **24**, (5), pp. 1433–1440
3 Bastos, J., Steyaert, M., Pergoot, A., Sansen, W.: 'Mismatch characterization of submicron MOS transistors', *Analog Integr. Circuits Signal Process.*, 1997, **12**, (2), pp. 95–106
4 Phelps, R., Krasnicki, M., Rutenbar, R.A., Carley, L.R., Hellums, J.R.: 'Anaconda: simulation-based synthesis of analog circuits viastochastic pattern search', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2000, **19**, (6), pp. 703–717
5 Sahu, B., Dutta, A.K.: 'A fuzzy logic based approach for parametric optimization of MOS operational amplifiers', *Microelectron. J.*, 2002, **33**, (3), pp. 253–264
6 Abdel-Malek, H.L., Hassan, A.K.S.O.: 'The ellipsoidal technique for design centering and region approximation', *IEEE Trans. Comput. Aided Des.*, 1991, **10**, (8), pp. 1006–1014
7 Antreich, K.J., Graeb, H.E., Wieser, C.U.: 'Circuit analysis and optimization driven by worst-case distance', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 1994, **13**, (1), pp. 57–71
8 Dharchoudhury, A., Kang, S.M.: 'Worst-case analysis and optimization of VLSI circuit performances', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 1995, **14**, (4), pp. 481–492
9 Mukherjee, T., Carley, L.R., Rutenbar, R.A.: 'Efficient handling of operating range and manufacturing line variations in analog cell synthesis', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2000, **19**, (8), pp. 825–839
10 Krishna, K., Director, S.W.: 'The linearized performance penalty (LPP) method for optimization of parametric yield and its reliability', *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 1995, **14**, (12), pp. 1557–1568
11 Antreich, K., Eckmueller, J., Graeb, H., *et al.*: 'WiCkeD: analog circuit synthesis incorporating mismatch'. Proc. Custom Integrated Circuits Conf., Orlando, USA, May 2000, pp. 511–514
12 Schwecker, R., Schenkel, F., Pronath, M., Graeb, H.: 'Analog circuit sizing using adaptive worst-case parameter sets'. Proc. Design, Automation and Test in Europe Conf. on Exhibition, Paris, France, March 2002, pp. 581–585
13 Roma, C., Daglio, P., Sandre, G.D., Pasotti, M., Poles, M.: 'How circuit analysis and yield optimization can be used to detect circuit limitations before silicon results'. Proc. Sixth Int. Symp. on Quality of Electronic Design, San Jose, USA, March 2005, pp. 107–112
14 Graeb, H., Zizala, S., Eckmueller, J., Antreich, K.: 'The sizing rules method for analog integrated circuit design'. Proc. Int. Conf. on Computer-Aided Design, San Jose, USA, November 2001, pp. 343–349
15 Schwecker, R., Schenkel, F., Graeb, H., Antreich, K.: 'The generalized boundary curve – a common method for automatic nominal design and

design centering of analog circuits'. Proc. Design, Automation and Test in Europe Conf. and Exhibition, Paris, France, March 2000, pp. 42–47
16 Lewis, R.M., Torczon, V., Trosset, M.W.: 'Direct search methods: then and now', *J. Comput. Appl. Math.*, 2000, **124**, (1), pp. 191–207
17 Graeb, H.E.: 'Analog design centering and sizing' (Springer, 2007)
18 Nocedal, J., Wright, S.J.: 'Numerical optimization' (Springer Science + Business Media, 2006)
19 Hooke, R., Jeeves, T.: 'Direct search solutions of numerical and statistical problems', *J. ACM*, 1961, **8**, (2), pp. 212–229
20 Kolda, T.G., Lewis, R.M., Torczon, V.: 'Stationarity results for generating set search for linearly constrained optimization', *SIAM J. Optim.*, 2006, **17**, (4), pp. 943–968
21 Bűrmen, Á., Strle, D., Bratkovič, F., Puhan, J., Fajfar, I., Tuma, T.: 'Automated robust design and optimization of integrated circuits by means of penalty functions', *Int. J. Electron. Commun.*, 2003, **57**, (1), pp. 47–56
22 Baker, R.J.: 'CMOS circuit design, layout, and simulation' (John Wiley & Sons, 2008)
23 Lasanen, K., Korkala, V., Raisanen-Ruotsalainen, E., Kostamovaara, J.: 'Design of a 1-V low power CMOS bandgap reference based on resistive subdivision'. Proc. 45th Midwest Symp. on Circuits and Systems, Tulsa, USA, August 2002, pp. 564–567
24 http://www.python.org, accessed March 2010
25 http://numpy.scipy.org/, accessed March 2010
26 Tuma, T., Bűrmen, Á.: 'Circuit simulation with SPICE OPUS, theory and practice' (Birkhaüser, 2009)
27 Olenšek, J., Bűrmen, Á., Puhan, J., Tuma, T.: 'DESA: a new hybrid global optimization method and its application to analog integrated circuit sizing', *J. Glob. Optim.*, 2009, **44**, (1), pp. 53–77
28 Box, M.J.: 'A new method of constrained optimization and comparison with other methods', *Comput. J.*, 1965, **8**, (1), pp. 42–52