Performance Trade-off Analysis of Analog Circuits By Normal-Boundary Intersection

Guido Stehr Helmut Graeb Kurt Antreich
Institute for Electronic Design Automation, Technical University of Munich, 80290 Munich, Germany
{stehr, graeb, antreich}@eda.ei.tum.de

ABSTRACT

We present a new technique to examine the trade-off regions of a circuit where its competing performances become "simultaneously optimal", i.e. Pareto optimal. It is based on circuit simulation, sizing rules, which capture elementary topological and technological constraints, and an advanced multicriteria optimization formulation called normal-boundary intersection. We are able to efficiently calculate a well-balanced discretization of a Pareto surface, identify the active constraints, which prevent a further improvement, and even rank these constraints in terms of stringency. Experimental results demonstrate the efficacy and efficiency of the method and its potential for topology selection and analog synthesis.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; B.7.2 [**Integrated Circuits**]: Design Aids—*simulation*; I.6.3 [**Simulation and Modeling**]: Applications; J.6 [**Computer-Aided Engineering**]: *Computer-Aided Design*

General Terms

Algorithms, Design, Performance

Keywords

analog circuits, Pareto optimality, normal-boundary intersection, trade-off analysis, performance space exploration, topology selection

1. INTRODUCTION

Analog components play important roles in modern integrated electronic systems. Signal conversion, clock generation, or data acquisition are just a few examples. The design of digital circuits takes great advantage of abstraction and therefore lends itself to automation. Analog components, in contrast, are heavily influenced by nonlinear low-level physical effects, which has been a great obstacle in their automated design. Therefore, in a system design, the analog part is often a bottleneck, and improved development aids are urgently needed.

Analog design consists mainly of three steps: topology selection, component sizing, and layout generation. In flat, as well as hierarchical design, in a first step, circuit topologies have to be chosen that are capable of satisfying given performance specifications. In a second step, actual values have to be assigned to the circuit parameters, primarily the transistor widths and lengths. It is not before this second step that the circuit performances such as DC gain of an operational amplifier can be simulated. Yet, already in the first step, a notion of the achievable circuit performances is necessary. This paper aims at a closer integration of these two design steps.

In order to obtain a "meaningful" sizing, designers consider numerous constraints, such as keeping transistors in saturation or matching them. These requirements, which are usually not explicitly given, are called *sizing rules*. Only that fraction of the design space, where all these rules are met, can be used for the optimization of the circuit performances.

There is usually not a unique design that optimizes the performances simultaneously. Instead, there is a trade-off situation, where it is only possible to improve one performance at the cost of another, which leads to the concept of *Pareto optimality* [1, 3]. In the sizing step, this conflict is usually resolved by assigning weights to the performances and combining them into a single optimization objective. However, it is hard to anticipate how the weights affect the selection of one particular solution. Therefore, knowledge of the entire set of "equally optimal" points , i.e. of the *Pareto optimal front*, would help to pick one distinct solution deliberately [8].

The calculation of trade-off curves has gained increased attention recently. The method presented in [4], which focuses on circuit sizing, relies on posynomial circuit models, which approximate the circuit performances and are extremely fast to evaluate. Therefore, a large number of sizings can easily be carried out in order to find trade-off curves and the limiting sizing rules. In contrast, the method of [7] is simulation-based but focuses on the numeric modeling of the Pareto surfaces rather than on their efficient determination. The simulation-based evolutionary algorithm presented in [2] results in high computational costs and cannot provide information about the limiting sizing rules.

The present approaches have the drawbacks that they either suffer from long simulation times and fail to classify the active sizing rules or require approximate behavioral circuit models. Therefore, in this paper, we suggest a technique that:

- works in a simulation-based environment for accuracy,
- keeps simulation costs low by a special optimization formulation for well-directed searches in the performance space, and
- gains valuable design information from the limiting sizing rules.

The remainder of the paper is structured as follows: In Sec. 2, we briefly summarize how sizing rules define the feasible parameter space. Our multi-objective optimization technique is explained in Sec. 3, whereupon in Sec. 4, we provide application examples. Sec. 5 concludes this paper.

2. FEASIBLE PARAMETER SPACE

Analog circuits are usually designed hierarchically: Individual transistors form transistor pairs which constitute elementary building blocks such as current mirrors or differential pairs. These transistor pairs are combined again to obtain larger building blocks such as cascode current mirrors. This combination of building blocks is continued until the design is completed. The final circuit has to satisfy all performance specifications, which are explicitly given and concern the entire circuit.

Yet, there are additional requirements on the basic building blocks, which can be interpreted as technological and topological design specifications. For example, a current mirror does not work properly unless its transistors operate in saturation. These requirements are usually not explicitly given but reflect design knowledge. For design automation purposes, however, these specifications have to be stated explicitly. This can be done by means of *sizing rules*, which are frequently mentioned in literature [4, 6, 9].

A systematic way to automatically set up these rules for a given circuit was presented in [6]: In a first step, the basic building blocks of a circuit are identified hierarchically based on a flat schematic. In a second step, generic sizing rules are instantiated and assigned to the actual transistors. Three categories suffice to fully classify a sizing rule:

1. Geometric / Electrical: Geometric sizing rules directly refer to transistor geometries. Electrical rules need to be evaluated based on circuit simulations.
2. Function / Robustness: Functional rules have to be met unconditionally in order to allow a building block to fulfill the desired function. Robustness rules account for global and local variations in the manufacturing process and the operating conditions.
3. Inequality / Equality: Inequality sizing rules require that electrical or geometric circuit quantities exceed or remain below certain limits. Equality rules exist only for geometric quantities. Since they are given as explicit algebraic equations, they can be used to reduce the dimension of the design space.

After the parameter space reduction on the basis of geometric equality sizing rules, the remaining parameters have to satisfy a number of inequalities, which are either explicitly given as algebraic expressions or require circuit simulation.

Beyond the sizing rules as described above, there are two more categories of constraints that further cut down the feasible parameter space: First of all, there are lower and upper bounds on every single geometric transistor parameter. The lower bounds are given by the employed CMOS technology. The upper ones can be chosen by the designer and make sure that no component becomes excessively large. Since these bounds on the different parameters are mutually independent, they define a hyperbox in the parameter space, and are therefore called *box constraints*. They guarantee that the feasible parameter space is a bounded region.

Furthermore, if a number of performances are to be optimized, there are usually specifications on the remaining performances. Although these upper and/or lower bounds obviously cannot be derived from a topological analysis of a circuit, they have the same appearance as electrical inequality sizing rules. Therefore, they can be treated in the same way mathematically.

In summary, the allowed parameter values $\mathbf{p}$ have to satisfy the sizing rules, the box constraints, and the performance specifications, which after elementary algebraic transformations can all be expressed as a vector inequality constraint $\mathbf{c}(\mathbf{p}) \geq \mathbf{0}$ [1]. Consequently, the feasible parameter space $\mathcal{P}$ can be written as

$$\mathcal{P} = \{\mathbf{p} \mid \mathbf{c}(\mathbf{p}) \geq \mathbf{0}\}\,. \tag{1}$$

[1]In this paper, regular lower case letters denote scalars. Vectors are written in bold lower case. Matrices are bold capitals.

A circuit sizing is considered feasible if and only if it satisfies (1). Note that $\mathcal{P}$ is given by the topology and the technology. This space defines the degrees of freedom for any performance optimization. Consequently, an accurate description of $\mathcal{P}$ is crucial to a trustworthy exploration of the performance space.

3. TRADE-OFF CALCULATION

3.1 Problem Formulation

Formally speaking, the circuit performances $\mathbf{f}$ are functions of the designable circuit parameters $\mathbf{p}$: $\mathbf{f}$=$\mathbf{f}(\mathbf{p})$. This nonlinear mapping can be evaluated using circuit simulations. The discussion of sizing rules showed that only a fraction of the entire parameter space leads to technically meaningful sizing results. As depicted in Fig. 1, the feasible parameter space $\mathcal{P} \subset \mathbb{R}^m$ has an image $\mathcal{F} \subset \mathbb{R}^n$ in the performance space with

$$\mathcal{F} = \{\mathbf{f} \mid \mathbf{f} = \mathbf{f}(\mathbf{p}) \wedge \mathbf{p} \in \mathcal{P}\}\,, \quad \mathcal{P} = \{\mathbf{p} \mid \mathbf{c}(\mathbf{p}) \geq \mathbf{0}\}\,. \tag{2}$$

In most cases, there are more parameters than performances, i.e. $m > n$. The boundary of $\mathcal{F}$ is denoted as $\partial\mathcal{F}$.

Usually, the design goal is to simultaneously obtain values for a number of circuit performances that can be considered optimal in some sense, while, at the same time, sizing rules, performance specifications, and box constraints have to be met. Such a multi-objective or multicriteria optimization problem can be stated as

$$\underset{\mathbf{p}}{\text{“optimize”}}\ \mathbf{f}(\mathbf{p}) = \begin{bmatrix} f_1(\mathbf{p}) \\ f_2(\mathbf{p}) \\ \vdots \\ f_n(\mathbf{p}) \end{bmatrix} \quad \text{s.t.} \quad \mathbf{c}(\mathbf{p}) \geq \mathbf{0}\,; \quad n \geq 2\,. \tag{3}$$

It is rarely possible to find a set of parameters that optimizes all performances at the same time. Instead, there is usually a trade-off situation where it is only possible to improve one performance at the cost of another. This leads to the concept of *Pareto optimality* [1, 3, 8].

Without loss of generality, in the remainder of this paper, optimization means minimization. In multi-objective optimization, a vector $\mathbf{a} = [a_1 a_2 \dots a_n]^T$ [2] is considered more optimal than a vector $\mathbf{b} = [b_1 b_2 \dots b_n]^T$ if it *dominates* $\mathbf{b}$:

$$\mathbf{a} \prec \mathbf{b} :\Leftrightarrow \underset{i\in\{1,2,\dots,n\}}{\forall} (a_i \leq b_i) \wedge \underset{i\in\{1,2,\dots,n\}}{\exists} (a_i < b_i)\,. \tag{4}$$

A vector $\mathbf{f}^*$ [3] is *Pareto optimal* within $\mathcal{F}$ if it is *nondominated*:

$$\neg \underset{\mathbf{f}\in\mathcal{F}}{\exists} \mathbf{f} \prec \mathbf{f}^*\,. \tag{5}$$

In Fig. 1 point $\mathbf{f}^\diamond$ is dominated, but all the points on the arc between $\mathbf{f}^{*a}$ and $\mathbf{f}^{*b}$ are nondominated. This portion of the boundary $\partial\mathcal{F}$ denoted as $\partial\mathcal{F}^\prec$ is called *Pareto optimal front*. This part of $\mathcal{F}$ is especially interesting to a designer since it characterizes the ultimate performance capability of a topology and the trade-offs involved. The points $\mathbf{f} \in \partial\mathcal{F}^\prec$ are also called *efficient* points. Our goal is to compute $\partial\mathcal{F}^\prec$ efficiently.

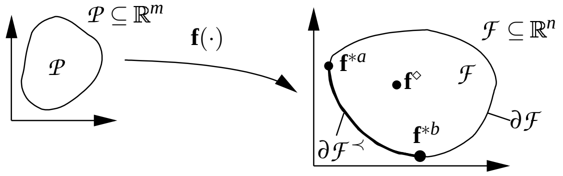

Figure 1: Feasible performance space $\mathcal{F}$ with Pareto optimal front $\partial\mathcal{F}^\prec$

[2]The superscript ^{T} identifies a transposed vector.
[3]Symbolic superscripts are used to mark special vectors.

3.2 Common Solution Methods

In practical problems, it is normally not possible to obtain a description of the entire Pareto optimal front in closed form. Instead, this region has to be approximated by a number of points. To calculate them, the multi-objective optimization problem (3) is usually transformed into a single-objective optimization formulation, which is then repeatedly solved.

3.2.1 Scalar Cost Function

One way to turn (3) into a single-objective optimization problem is to combine all objectives into one by means of a scalar cost function $s : \mathbb{R}^n \mapsto \mathbb{R}$. In the simplest and most common implementation of this idea s is a weighted sum:

$$\min_{\mathbf{p}} s(\mathbf{f}(\mathbf{p})) = \mathbf{w}^T \cdot \mathbf{f}(\mathbf{p}) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{p}) \geq \mathbf{0} ;$$
$$\mathbf{w} = [w_1 w_2 \ldots w_n]^T , \qquad \underset{i \in \{1,2,\ldots,n\}}{\forall} w_i > 0 . \qquad (6)$$

This approach is illustrated in Fig. 2, where two contour lines of $s(\mathbf{f})$ are shown. The solution $\mathbf{f}^*$ is where such a line is tangential to $\partial\mathcal{F}^{\prec}$. The orientation of the contour lines is determined by the weight vector $\mathbf{w}$. It can be shown that, for a convex Pareto optimal front, every efficient point is the solution of a problem according to (6) with a proper weight vector $\mathbf{w}$.

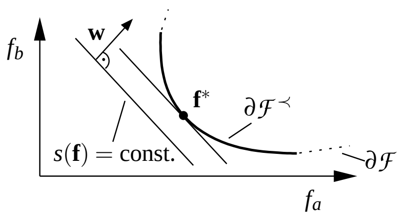
f_b
$\mathbf{w}$
$\mathbf{f}^*$
$\partial\mathcal{F}^{\prec}$
$s(\mathbf{f}) = \text{const.}$
$\partial\mathcal{F}$
f_a

Figure 2: Weighted sum: efficient point $\mathbf{f}^*$ according to (6)

Yet, the weighted sum method has two major drawbacks. As shown in the left part of Fig. 3, an even distribution of the weight vector $\mathbf{w}$ does not yield evenly spread Pareto points. Instead, they cluster in regions with strong curvature, which are typical of ill-conditioned technical problems. Additionally, the method is not able to detect all points on nonconvex Pareto fronts. In the right part of Fig. 3, the points on the arc between $\mathbf{f}^{\diamond}$ and $\mathbf{f}^{\square}$ would not be obtainable using the weighted sum method. Other approaches employing a scalar cost function basically have the same problems.

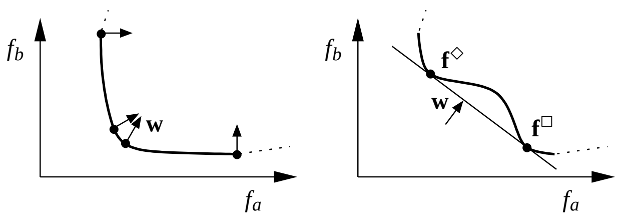

Figure 3: Weighted sum drawbacks: clustering and nondetectable efficient points in nonconvex regions

3.2.2 Objectives as Constraints

A different method to obtain a scalar optimization problem is to turn all objectives but one into constraints [7]:

$$\min_{\mathbf{p}} f_j(\mathbf{p}) \quad \text{s.t.} \quad \underset{i \in \{1,2,\ldots,n\} \setminus \{j\}}{\forall} f_i \leq f_{i,max} \ \wedge \ \mathbf{c}(\mathbf{p}) \geq \mathbf{0} ;$$
$$j \in \{1,2,\ldots,n\} . \qquad (7)$$

Here, the optimization is controlled by varying the maximum values of the n–1 performance constraints $f_{i,max}$. In contrast to the weighted sum method, this approach also yields the efficient points in nonconvex parts of the Pareto optimal front as shown in Fig. 4, left. In fact, the constraint method does not yield any dominated points. In Fig. 4, right, an optimization with $f_{i,max3}$ as bound would yield $\mathbf{f}^{\diamond}$. Therefore, the arc between $\mathbf{f}^{\diamond}$ and $\mathbf{f}^{\square}$ would result in a gap in the solution curve. Although in the mathematical sense this is a merit of the constraint method, a designer might be interested also in these non-Pareto parts of $\partial\mathcal{F}$.

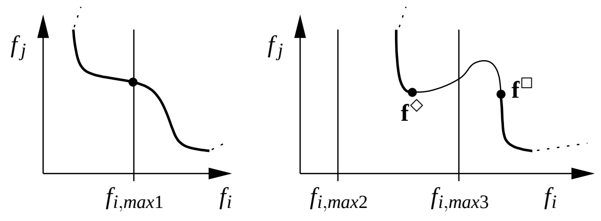

Figure 4: Constraint method detects efficient points in nonconvex regions and leaves out non-efficient parts

In contrast to the weighted sum formulation, which yields solutions for any weight vector with non-negative entries, the constraint method does not lead to solutions if the specified bounds are infeasible as shown with $f_{i,max2}$ in Fig. 4, right. On the other hand, different bounds may lead to the same Pareto point.

While the constraint method overcomes the limitation regarding nonconvex Pareto curves, it still does not generate evenly distributed points on the Pareto front. Therefore, we introduce the normal-boundary intersection approach [1] to the area of circuit design. This technique goes even further than the constraint method in that it transforms *all* objectives into constraints.

3.3 Normal-Boundary Intersection

3.3.1 General Idea

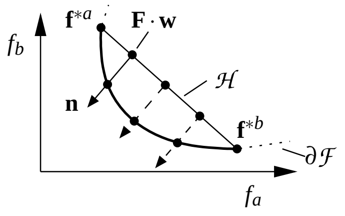

Figure 5: Normal-boundary intersection

The normal-boundary intersection method (NBI) relies on four corner stones as illustrated in Fig. 5:

1. individual minima: $\mathbf{f}^{*i}$
   There are special points $\mathbf{f}^{*i}$ in a Pareto front where the individual performances f_i show their global minima. These points are boundary points of the Pareto front and are called *individual minima*. It has been pointed out in the previous section that caution has to be taken not to initiate searches in regions where no efficient points can be found. Therefore, in a first step, extreme points of the Pareto front are sought. If no two performances can be minimized at the same time, there is one unique individual minimum for each of the n performances. Beyond their algorithmic importance, the individual minima are of great interest to a designer.
2. convex hull of individual minima: $\mathcal{H}$
   The basic idea behind NBI is that there are Pareto optimal points "underneath" the convex hull $\mathcal{H}$ of the individual minima. Geometrically, $\mathcal{H}$ is a polyhedron with the individual minima as its corner points. It is (n–1)-dimensional if n individual minima exist.

3. normal vector to convex hull: **n**
A search on a line perpendicular to $\mathcal{H}$ towards the origin yields Pareto optimal points if the respective portion of $\partial\mathcal{F}$ is convex. If the searches start from points evenly spaced within $\mathcal{H}$ and run in the direction of a vector **n** normal to $\mathcal{H}$, the solution points on the efficient front are also well-balanced. Note that it is not even necessary for **n** to be exactly orthogonal to $\mathcal{H}$. The only requirement on **n** is that it contains significant negative entries for every performance f_i.

4. NBI optimization problem formulation
A suitable optimization formulation captures the geometric deliberations from above mathematically. The NBI optimization problem can be solved using nonlinear optimization.

3.3.2 Algorithmic Details

In this section, the four NBI main ideas from above are elaborated mathematically.

1. individual minima: $\mathbf{f}^{*i}$
The individual minima $\mathbf{f}^{*i}$ can be determined using a conventional single-objective optimization formulation in f_i:

$$\mathbf{p}^{*i} = \underset{\mathbf{p}}{\operatorname{argmin}}\ f_i(\mathbf{p}) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{p}) \geq \mathbf{0}\,;$$
$$\mathbf{f} = [f_1 f_2 \ldots f_i \ldots f_n]^T\,, \quad \mathbf{f}^{*i} = \mathbf{f}(\mathbf{p}^{*i})\,. \qquad (8)$$

The argmin operation yields the argument leading to the minimum objective value. Let

$$f_{i,min} = f_i^{*i} \qquad (9)$$

denote the optimal value of performance f_i as obtained from (8). Analogously, $f_{i,max}$ could be defined as the upper bound of performance f_i within $\partial\mathcal{F}^{\prec}$. Since the exact knowledge of these bounds is of minor interest, it is sufficient to estimate them by the largest objective values occurring in the set of individual minima $\{\mathbf{f}^{*1}, \mathbf{f}^{*2}, \ldots, \mathbf{f}^{*n}\}$:

$$f_{i,\widetilde{max}} = \max(f_i^{*1}, f_i^{*2}, \ldots, f_i^{*n})\,, \quad i \in \{1,2,\ldots,n\}\,. \qquad (10)$$

Adequate normalization of the objective values is crucial to any numeric optimization. We use normalized performance values according to (9), (10) and

$$\hat{f}_i = \frac{(f_i - f_{i,min})}{(f_{i,\widetilde{max}} - f_{i,min})}\,, \quad i \in \{1,2,\ldots,n\}\,. \qquad (11)$$

This normalization maps the performance values to a range $[0,u]$. Here, $u = 1$ if (10) yields the true maximum values.

2. convex hull of individual minima: $\mathcal{H}$
If the normalized individual minima are combined in a matrix

$$\mathbf{F} = \left[\hat{\mathbf{f}}^{*1}\, \hat{\mathbf{f}}^{*2} \ldots \hat{\mathbf{f}}^{*n}\right]\,, \qquad (12)$$

then their convex hull can be written as

$$\mathcal{H} = \mathbf{F}\cdot\mathbf{w}\,, \quad \mathbf{w} = [w_1 w_2 \ldots w_n]^T\,,$$
$$w_i \geq 0\,, \quad \sum_i w_i = 1\,, \quad i \in \{1,2,\ldots,n\}\,. \qquad (13)$$

Note that the diagonal elements of **F** are all zeros due to the normalization (11).

3. normal vector to convex hull: **n**
The search along a family of normal vectors **n** offers the benefit of well-balanced solution points. This trait is only marginally impaired if the search direction is not exactly normal to $\mathcal{H}$. Therefore, the following quasi-normal vector $\tilde{\mathbf{n}}$, which is very easy to calculate, is sufficient:

$$\tilde{\mathbf{n}} = -\mathbf{F}\cdot\mathbf{1}\,, \quad \mathbf{1} = [11\ldots1]^T\,. \qquad (14)$$

This is illustrated in Fig. 6. The dotted arrows show the negative vector sum of the individual minima $\mathbf{f}^{*a}$ and $\mathbf{f}^{*b}$ which yields vector $\tilde{\mathbf{n}}$. It is evident that a search along this quasi-normal direction also yields good results.

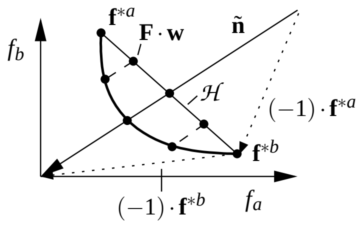
$\mathbf{f}^{*a}$
$\mathbf{F}\cdot\mathbf{w}$
$\tilde{\mathbf{n}}$
f_b
$\mathcal{H}$
$(-1)\cdot\mathbf{f}^{*a}$
$\mathbf{f}^{*b}$
$(-1)\cdot\mathbf{f}^{*b}$
f_a

Figure 6: NBI with quasi-normal vector ñ (not normalized)

4. NBI optimization problem formulation
Finally, the following NBI optimization formulation combines all components described above:

$$\begin{bmatrix}\mathbf{p}^*\\ t^*\end{bmatrix} = \underset{\left[\begin{smallmatrix}\mathbf{p}\\ t\end{smallmatrix}\right]}{\operatorname{argmax}}\ t \quad \text{s.t.} \quad \mathbf{F}\cdot\mathbf{w} + t\cdot\tilde{\mathbf{n}} = \hat{\mathbf{f}}(\mathbf{p}) \ \wedge\ \mathbf{c}(\mathbf{p}) \geq \mathbf{0}\,;$$
$$w_i \geq 0\,, \quad \sum_i w_i = 1\,, \quad i \in \{1,2,\ldots,n\}\,;$$
$$\mathbf{f}^* = \mathbf{f}(\mathbf{p}^*)\,. \qquad (15)$$

For a given vector **w**, the solution to this problem is a unique Pareto optimal point $\mathbf{f}^*$. Note that the objective function just consists of a newly introduced parameter t. The geometric idea behind NBI is coded in a vector equality constraint: The term $\mathbf{F}\cdot\mathbf{w} \in \mathcal{H}$ defines the base point of the quasi-normal vector $\tilde{\mathbf{n}}$. Therefore, the left-hand side of the equality constraint describes all points along the quasi-normal. The parameter t is a measure of the distance from $\mathcal{H}$. Since $\mathbf{c}(\mathbf{p}) \geq \mathbf{0}$ ensures the feasibility of the parameter vectors, $\hat{\mathbf{f}}(\mathbf{p}) \in \mathcal{F}$ is a feasible performance vector according to (2). This optimization formulation can be seen as a *line search in the performance space*: The goal is to find a normalized feasible performance vector that has the maximum distance from $\mathcal{H}$.
For an even spread of k efficient points, the components of **w** can be chosen according to

$$w_i = \frac{j_i}{k}\,; \quad k \in \mathbb{N}^+\,,$$
$$j_i \in \begin{cases} \{0,1,\ldots,k\cdot(1-\sum_{l=1}^{i-1} w_l)\}\,, & 1 \leq i < n \\ \{k\cdot(1-\sum_{l=1}^{i-1} w_l)\}\,, & i = n \end{cases}\,. \qquad (16)$$

Prior to practical implementations, the conceptual parameter t can be eliminated from the problem (15) algebraically. For the two-dimensional case, (15) can be simplified to

$$\min_{\mathbf{p}}\ \hat{f}_1 \quad \text{s.t.} \quad \hat{f}_1 - \hat{f}_2 + 2w - 1 = 0 \quad \wedge \quad \mathbf{c}(\mathbf{p}) \leq \mathbf{0}\,;$$
$$0 < w < 1\,. \qquad (17)$$

Owing to the normalization given in (11), vector $\tilde{\mathbf{n}}$ from (14) turns out exactly normal in this case: $\tilde{\mathbf{n}} = [-1-1]^T$, see Fig. 7.

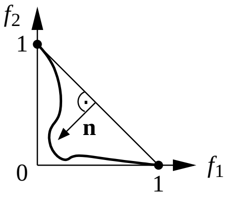

Figure 7: Normalized NBI in 2 dimensions

This figure also shows that NBI can trace mildly nonconvex boundaries as well. In contrast to path following methods, NBI is applicable to higher dimensions as well. Additionally, the pre-image of the efficient front, i.e. the set of points which are mapped onto $\partial\mathcal{F}^{\prec}$, does not have to be continuous.

3.3.3 Practical Solution via SQP

To operationally solve the NBI problem, we use a state-of-the-art SQP algorithm in combination with a SPICE-like circuit simulator.

For every sample point on $\partial\mathcal{F}^{\prec}$, one optimization run has to be carried out. Often, a new solution vector can be found in the neighborhood of the previous solution. Therefore, we reuse the approximation of the Hessian matrix and the solution vector found in one optimization run to jump-start the SQP algorithm for the the subsequent optimization.

For a shorter execution time, our approach can be parallelized: In the first step, the individual minima can be calculated in parallel according to (8). In the second step, the remaining efficient points are calculated using (15), which can be done in parallel. In order to retain the jump start feature, however, it might be preferable to proceed from the individual minima towards the center of $\mathcal{H}$.

We are aware of the problem that, like any gradient-based technique, SQP can get trapped in local optima. This might yield solution points which are mistakenly classified as Pareto optimal. However, it was observed that circuit performances are usually well-natured as soon as all sizing rules are fulfilled [2, 6].

While evolutionary approaches [2] only yield solution points more or less close to the Pareto front, a deterministic search advances to the actual border of $\mathcal{P}$. The advantage is two-fold: First of all, the solution points are really located *on* the boundary of $\mathcal{P}$. Additionally, the SQP algorithm identifies the entire set of active, i.e. performance-limiting sizing constraints, and can even rank them in terms of stringency as explained below.

Since the vector equality constraint of the NBI problem (15) does not correspond to any sizing rule, it can be neglected in this context. For a total number of r inequality constraints, the Lagrangian function is

$$L(\mathbf{p},\boldsymbol{\lambda}) = t - \sum_{j\in\{1,2,\ldots,r\}} \lambda_j \, c_j(\mathbf{p}) \,. \qquad (18)$$

In a solution point, some of the constraints are active, i.e. $c_j(\mathbf{p}) = 0$. This set of active constraints reveals "design bottlenecks" because the circuit performances cannot further be improved without violating the corresponding sizing rules, box constraints, or performance specifications.

By convention, $\lambda_j \neq 0$ only for the active constraints. If the right-hand sides of the active constraints are disturbed, i.e. $c_j(\mathbf{p}) = \varepsilon_j$, then the optimum value t^* of the objective changes accordingly. It is well-known [5] that the Lagrange multipliers can be interpreted as the sensitivities of the optimum objective values with respect to variations of the active constraints:

$$\frac{\mathrm{d}t^*}{\mathrm{d}\varepsilon_j} = \lambda_j \,. \qquad (19)$$

From (15) it can be derived that

$$\mathrm{d}\hat{\mathbf{f}}^* = \tilde{\mathbf{n}}\,\mathrm{d}t \,. \qquad (20)$$

Substitution and denormalization (cf. (11)) yield the sensitivities of the circuit performances

$$\frac{\mathrm{d}f_i^*}{\mathrm{d}\varepsilon_j} = (f_{i,\widetilde{max}} - f_{i,min}) \cdot \tilde{n}_i \cdot \lambda_j \,. \qquad (21)$$

These sensitivities are of great importance to a designer because they are a measure of how stringent the active constraints are in a certain Pareto point.

4. APPLICATIONS AND RESULTS

4.1 Performance Space Exploration

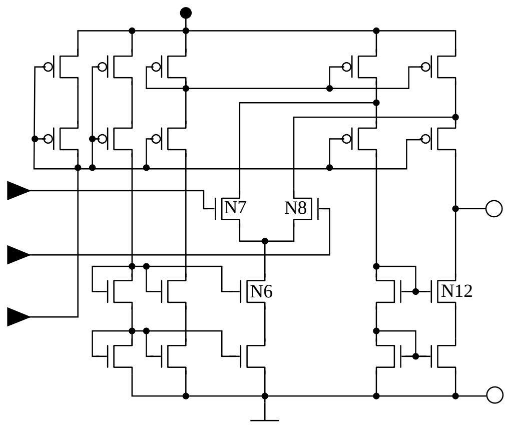

Figure 8: Folded cascode operational amplifier

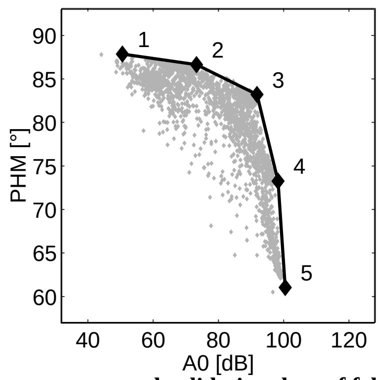

Figure 9: Pareto curve and validation data of folded cascode

As an example, the folded cascode architecture given in Fig. 8 was analyzed using the NBI method based on a 0.65 μm process. The Pareto optimal front of the two performances DC gain (A0) and phase margin (PHM) is depicted in Fig. 9. The amplifier is subject to 204 inequality sizing rules, and, in this example, the remaining performances are unconstrained. Since the goal is to maximize both performances, each of them has to be multiplied by –1 in order to obtain a minimization problem according to (8)–(15). In this sense, points 1 and 5 are the individual minima of DC gain and phase margin, respectively (cf. (8)). Three NBI runs are performed according to (15) in order to obtain points 2 to 4. Interpolation yields an approximation of the entire Pareto front. The result was validated by extensive simulation runs, as indicated by the gray dots.

It is evident that this operational amplifier cannot achieve a DC gain substantially exceeding 100 dB. This maximum gain leads to a phase margin of roughly 62°. The trade-off curve shows that "the last few dBs" come at the price of significantly lowering the phase margin. In the mathematical sense, all efficient points are equally optimal. Yet, a designer might choose point 3 as "best" point because it combines a "relatively high" gain with a "relatively high" gain margin. This point, where the efficient front has the strongest curvature, is called the "knee" of the Pareto curve.

4.2 Examination of Limiting Sizing Rules

In addition to the ultimate performance capabilities of a circuit topology, our approach reveals the obstacles of a further improvement. In the example at hand, a number of 14 sizing rules turn out

sizing rule	Pareto points 1	2	3	4	5
lowerBound_w(N7,N8)	●	●	●		
minimumVds(N6)				●	●
minimumVgs(N12)					●
⋮			⋮		

Table 1: Active sizing rules

to have a limiting effect on the performances. Tab. 1 focuses on three of them. The spots indicate where the different sizing rules are active. It can be seen that the phase margin is limited by the minimum width of the transistors N7 and N8 (points 1–3). The DC gain, on the other hand, is restricted among others by the minimum drain-source voltage of N6 (points 4,5). The associated sizing rule is meant to ensure saturation and is given by

$$v_{ds} - (v_{gs} - V_{th}) \geq V_{sat,min}\,. \tag{22}$$

In this formula, v_{ds} denotes the drain-source voltage, v_{gs} the gate-source voltage and V_{th} the threshold voltage. The safety margin $V_{sat,min}$ is chosen by the designer. According to Tab. 1, lowering this margin would result in a higher DC gain. An analysis of the respective Lagrange multiplier at point 5 yields a sensitivity of

$$\frac{\mathrm{d}A0}{\mathrm{d}V_{sat,min}} = -19\ \frac{\mathrm{dB}}{\mathrm{V}}\,. \tag{23}$$

In this example, we chose $V_{sat,min} = 0.1$V. According to the linear estimation, giving up this margin entirely would only yield a gain improvement of about 2 dB. Therefore, it might be preferable to ensure saturation of N6 by a topological modification.

In this way, valuable design information can be gained by examining the set of active design rules and the associated sensitivities.

4.3 Topology Selection

In applications, where a certain performance trade-off (e.g. speed vs. power or amplification vs. phase margin) is critical, and specifications are given for the remaining performances, Pareto curves are a valuable tool for topology selection.

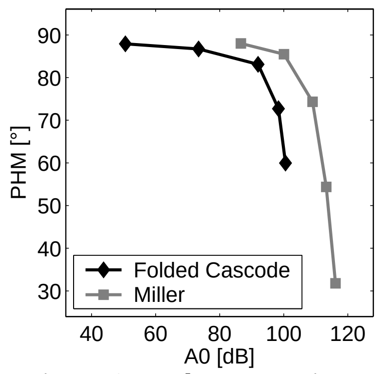

Figure 10: topology comparison

Fig. 10 compares performances of the folded cascode amplifier at hand and a Miller compensated opamp. Although the trade-offs are similar qualitatively, the Miller amplifier turns out superior in terms of DC gain and phase margin.

4.4 Computational Effort

amplifier	runtime [min:sec]	# simulations
folded cascode	10:29	962
Miller	13:12	1019

Table 2: Computational effort

Tab. 2 summarizes the computational effort it took to calculate the above two Pareto curves on a cluster of Pentium III machines. The Pareto points were calculated sequentially, and the gradients for the SQP algorithm were calculated in parallel using finite differences. While [2] reports simulation times of several hours, our approach yields the requested Pareto curves within minutes, which makes our method applicable to practical circuit design and to topology selection in analog synthesis. Our technique is well-suited for cell-level design using circuit simulation and for system-level design using adequate performance models.

5. CONCLUSIONS

In this paper, we describe a new simulation-based approach to explore the Pareto optimal regions of a circuit performance space. While sizing rules ensure a technically meaningful circuit sizing, a line search in the performance space yields the desired trade-off points. Our optimization formulation overcomes several limitations of alternative approaches and enables a good approximation of Pareto optimal regions with a low number of sample points.

With two operational amplifiers as examples we demonstrated how our method can be used to explore the ultimate performance capabilities of a given circuit topology at full transistor-level accuracy. The visualization of the results helps to examine the trade-offs between competing performances and to compare different topologies. Owing to a gradient-based optimization algorithm it is now possible in a simulation-based environment to identify the limiting sizing rules and quantify their stringency.

6. ACKNOWLEDGMENTS

The authors would like to thank Dr. M. Ulbrich, University of Hamburg, for helpful advice.

7. REFERENCES

[1] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, Aug. 1998.

[2] B. De Smedt and G. G. E. Gielen. WATSON: Design space boundary exploration and model generation for analog and RF IC design. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 22(2):213–223, Feb. 2003.

[3] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience Series in Systems and Optimization. Wiley, 2001.

[4] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee. Optimal design of a CMOS Op-Amp via geometric programming. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 20(1):1–21, Jan. 2001.

[5] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 1987.

[6] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The sizing rules method for analog integrated circuit design. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 343–349, 2001.

[7] R. Harjani and J. Shao. Feasibility and performance region modeling of analog and digital circuits. *Analog Integrated Circuits and Signal Processing*, 10(1):23–43, June 1996.

[8] M. Lightner, T. Trick, and R. Zug. Circuit optimization and design. *Circuit Analysis, Simulation and Design, Part 2 (A. Ruehli). Advances in CAD for VLSI 3*, pages 333–391, 1987.

[9] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts. AMGIE–A synthesis environment for CMOS analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 20(9):1037–1058, Sept. 2001.