

Efficient Approximation of Performance Spaces for Analog Circuits via Multi-Objective Optimization

Benedikt Ohse, David Schreiber, Jürgen Kampe, Christopher Schneider
 Ernst-Abbe-Hochschule Jena, University of Applied Sciences – Jena, Germany
 {benedikt.ohse, david.schreiber, juergen.kampe, christopher.schneider}@eah-jena.de

Abstract—This paper presents an adaptation of the well-known normal boundary intersection (NBI) method for approximating complete feasible performance spaces of analog integrated circuits. Those spaces provide accurate information about all feasible combinations of competing performance parameters in a circuit. While the NBI-method is originally designed for computing the so-called Pareto front of a multi-objective optimization problem only, it can be adapted for approximating the complete performance space with some modifications. A scalarization into single-objective optimization problems is performed within our developed tool, which can be connected to any Spice-based simulator. Besides presenting the algorithm and its adaptations, the focus lies on investigating parallelization techniques and their effect on decreasing the computational time. Numerical experiments show the computed approximations of two- and three-dimensional performance spaces of several OTAs and compare the efficiencies of different parallelization schemes.

Index Terms—analog circuits, performance space, optimization

I. INTRODUCTION

Multi-objective optimization strategies have become increasingly important and useful for the practical design of integrated analog circuits in the last decades. Not only do they allow to simplify and accelerate the design process, but also do they make design decisions less dependent on the experience, which is necessary to satisfy the idea of an analog synthesis approach [1]. Various methods and software tools are available for automated numerical sizing, centering, and modeling of circuits, e.g. [2]. With *Pareto optimization* [3], we will provide a suitable method for modeling the complete performance space as a tool for initial decisions for analog system designers or automated synthesis approaches.

In analog circuits, the *performance parameters* (e.g. bandwidth, power consumption, area consumption) typically compete with each other. Therefore, an *optimal compromise* for those criteria is aimed for. However, there is not one combination of *design parameters*, also called *utopian point*, that optimally satisfies all performance criteria simultaneously. Therefore, a set is considered, where no point is dominated by another. This set is called *Pareto front*. The *normal boundary intersection* (NBI) method [4] is a reliable deterministic method for approximating this set. For each Pareto point, a circuit sizing process must be performed. In [5], NBI is used for

approximating the Pareto front with gradient-based optimization methods. The successive NBI-method [6] establishes an improved iterative approach to generate the entire Pareto front including its boundary regions. Further modifications to the NBI-method were presented in [7]–[9]. Besides gradient-based methods, also stochastic optimization strategies were applied in the literature, like simulated annealing [10], particle swarm optimization [11], [12], or evolutionary approaches [13], [14]. Furthermore, there are combinations of both gradient-based solvers and stochastic solvers, e.g. Bayesian optimization [15].

This work focuses on the description, adaptation, and application of the NBI-method for approximating the *complete feasible performance space* (CFPS), which provides a presentation of the set of all attainable performance parameter combinations, compare Fig. 1. Complete performance space models are well-suited for comparing circuit topologies and creating efficient high-level simulation models. Such models are very useful for analog and mixed-signal designers of systems-on-chip, especially while making topology decisions. Moreover, these models can support efficient system-level simulations, since they deliver information about circuit property boundaries with the accuracy of a transistor-level simulation. With this, also a topology proposal that meets the specification for the system-level task can be given to the analog designer. Even if design experience is available, the presented models can support the analog designer with a visualization of the influence of technical restrictions [16]. This can contribute to a better objective evaluation of the circuit and facilitate the decision to switch to topologies with lower complexity. In addition, these models can be useful for analog synthesis algorithms to perform the initial topology selection step.

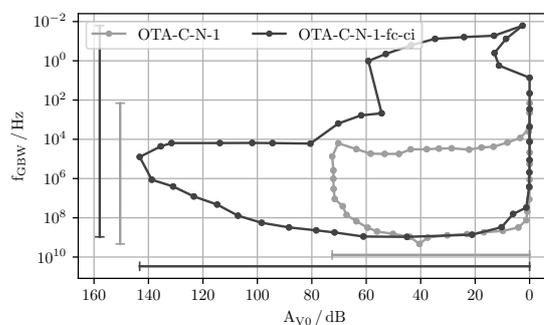


Fig. 1. Two-dimensional performance spaces of OTAs with different transistor-level topologies, computed automatically by the presented tool.

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) within the research project EGEVATIS under grant no. 03FH014X6 and by the graduate fellowship of Thuringia in Germany.

The Complete Feasible Performance Space

The goal of this work is the approximation of the entire performance space. Compared to the Pareto front, the effort increases exponentially, e.g. fourfold for the two-dimensional case and eightfold for the three-dimensional case. While multi-objective optimization typically aims for computing or approximating the Pareto front, there are good reasons for computing the CFPS:

(1) Analog integrated circuits are never designed as stand-alone components and always used in a system. Therefore, it does not make sense to design, e.g., an amplifier for maximum gain, as this would affect the stability of the system. In addition, process variations during chip fabrication lead to technology-dependent deviations from nominal performance values. Therefore, small perturbations may lead to infeasible circuit designs. The logical consequence is that a *safety margin* to the Pareto front should be maintained. For this, however, the knowledge about the shape of the region dominated by the Pareto front is important.

(2) To make reliable topology decisions at early design stages, CFPS-supported system-level simulations can help [17], e.g., to decide whether a topology of an analog circuit part is suitable for the application and to match system components to each other. Therefore, we need to know what properties can be achieved with each component. To create such simulation models (e.g. in SystemC, Matlab), the Pareto front is not sufficient, because a description of the limitation of all other feasible circuits is needed.

In Section II, the multi-objective optimization problem for computing the CFPS is defined. Then, in Section III, the focus lies on the here used, modified NBI-method, which is explained step by step. Section IV then presents numerical results for several OTAs as well as a discussion on the efficiency of different parallelization schemes. A conclusion is given in Section V.

II. THE OPTIMIZATION PROBLEM

All matrices are denoted by capital letters, e.g. H , to distinguish them from scalars t or vectors x . All inequality operators \leq and \geq are used component-wise for vectors. Sets are written in calligraphic letters, e.g. \mathcal{D} .

A. Multi-objective optimization

The adjustable *design parameters* of an analog circuit are denoted by $x \in \mathbb{R}^n$. These are, e.g., the channel length and width of transistors or the dimensions of passive components such as resistors and capacitors. They are constrained by inequalities $c(x) \geq 0$, mostly resulting from design rules of the semiconductor fabrication process, but also from constraints ensuring the electrical functionality of a circuit. Lower and upper technical limitations are considered as bounds $x_\ell \in \mathbb{R}^n$ and $x_u \in \mathbb{R}^n$, resp. [18] gives suggestions on reasonable constraints for the sizing of analog circuits, which can be determined by operating point simulations. The *design parameter space* $\mathcal{D} \subseteq \mathbb{R}^n$ includes all parameters, that satisfy the constraints:

$$\mathcal{D} = \{x \in \mathbb{R}^n \mid c(x) \geq 0, x_\ell \leq x \leq x_u\} \quad (\text{D})$$

We are interested in optimizing the performances $f_i(x)$ of a circuit which are, e.g., gain, bandwidth, power consumption and area consumption. These are summarized in the vector-valued objective function $F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^\top$. All $f_i(x)$ are chosen such that the optimal value is obtained by minimization. This can be achieved by a multiplication with -1 , if necessary. For all $x \in \mathcal{D}$, the attainable objective values $F(x)$ form the *performance space* $\mathcal{P} \subseteq \mathbb{R}^m$:

$$\mathcal{P} = \{F(x) \in \mathbb{R}^m \mid x \in \mathcal{D}\} \quad (\text{P})$$

Typically in multi-objective optimization (MOP), one would like to compute the Pareto front (all non-dominated solutions) by solving the following optimization problem:

$$\min_{x \in \mathcal{D}} F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^\top \quad (\text{MOP})$$

But as explained before, we aim for computing—more precisely *approximating*—the entire image space \mathcal{P} , since it represents the CFPS of the circuit. As we will see in the following, Problem (MOP) is also helpful for this task.

B. Scalarization of MOP

The NBI-method [4] is a numerical approach for approximating the Pareto front of multi-objective optimization problems like (MOP). Therby, among other steps, (MOP) is decomposed into several single-objective optimization problems (GA) like in [19]. This method uses so-called *base points* b —that can lie either inside or outside the performance space \mathcal{P} —which are defined as product of a matrix H and a weight vector $w \in \mathbb{R}^m$. The matrix H consists of two or more of the individual minima of each $f_i(x)$ [4]. The vector w consists of m elements which add up to one. A vector v indicates the *direction* of the optimization starting from the base point b . This is done by minimizing an additional scalar parameter t . Thus, a scalar optimization problem is formed:

$$\min_{x \in \mathcal{D}, t \in \mathbb{R}} t \quad \text{s.t.} \quad F(x) \leq b + tv, \quad b = Hw \quad (\text{GA})$$

While the original version contains equality constraints, as suggested in [20], the goal-attainment (GA) method [21] provides an improved approach. Here, the constraints become inequality constraints. With decreasing t , the intersection of $F(x)$ and the constraint becomes smaller until a solution point x on the Pareto front is reached. A geometric visualization of this can be seen, e.g., in [19].

To generate not only the Pareto front but the CFPS with the help of scalarized problems of type (GA), we use a rotation scheme for the constraints as introduced in [19]. Now, by using a set of base points b with suitable directions v , the entire boundary of the performance space \mathcal{P} can be approximated. This procedure is explained in detail in the following section.

III. CFPS APPROXIMATION APPROACH

A. Initialization

Each performance criterion $f_i(x)$ is individually minimized and maximized to obtain the vertices for an *initial polyhedral approximation* of the CFPS:

$$\min_{x \in \mathcal{D}} \pm f_i(x) \quad (1)$$

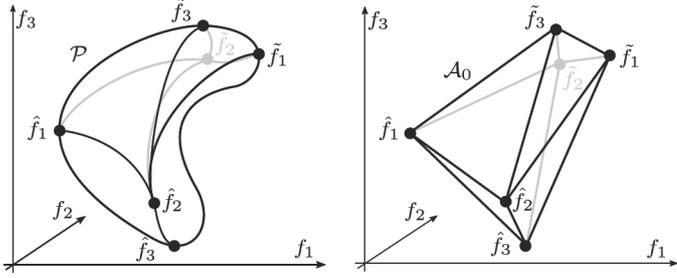


Fig. 2. Image points of individual minima and maxima (left) and initial polyhedral approximation of the performance space (right).

We denote the image points belonging to those individual minima and maxima with \hat{f}_i and \tilde{f}_i , resp. After a normalization, the convex hull of these points yields a first approximation of the performance space \mathcal{P} , compare Fig. 2:

$$\mathcal{A}_0 = \text{conv} \left\{ \hat{f}_1, \tilde{f}_1, \dots, \hat{f}_m, \tilde{f}_m \right\}$$

For implementation, the individual minima and maxima get renamed and numbered: $\bar{f}_1, \dots, \bar{f}_{2m}$ (compare Fig. 3–5).

B. Subproblem Definition

For a successive refinement of the CFPS-approximation, subproblems of different dimensions have to be defined and solved (Sec. III-E). Those subproblems are described by the convex hull \mathcal{S} of the involved vertices and the set of their associated indices \mathcal{I} :

$$\mathcal{I}_{k,j}, \quad \mathcal{S}_{k,j} = \text{conv} \left\{ \bar{f}_i \mid i \in \mathcal{I}_{k,j} \right\}, \quad k = 1, \dots, m-1$$

All $\mathcal{S}_{k,j}$ are k -dimensional faces (vertices, edges, faces, ...) of the initial convex hull \mathcal{A}_0 , where $j \in \mathbb{N}$ is the numerator. The subproblems are defined successively, starting with the highest dimension of the faces of \mathcal{A}_0 , as detailed in Fig. 3.

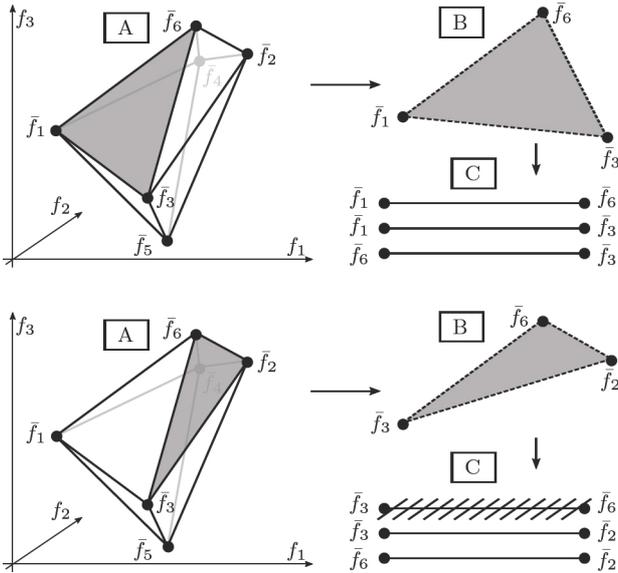


Fig. 3. Subproblem generation: choose face of highest dimension (A), build subproblem (B), partition into lower dimensional subproblems without doubles (C).

C. Base Point Distribution

The quality of the generated approximation highly depends on the distribution of base points b on \mathcal{A}_0 . Each base point will be used to compute a solution point on the boundary $\partial\mathcal{P}$ of the CFPS by solving the scalarized problem (GA) (see Sec. II-B). We start by distributing base points on each edge of \mathcal{A}_0 (subproblems of dimension $k = 1$). Depending on the length of the edge, the number of set base points varies. In [6], a linear program is now proposed to generate evenly distributed base points on the faces of higher dimension $k \geq 2$. While this is an efficient method, it can only be used if the number of base points on each edge is constant. Therefore, we use a modified *centroidal Voronoi tessellation* (CVT), cf. [22].

D. Search Direction

For solving the scalarized problems (GA), next to base points b , also suitable search directions v are needed. The original NBI-method [4] therefore uses quasi-normal vectors of the associated faces. But this approach only works for approximating the Pareto front (compare also [20]). Instead, we compute *normal vectors* for all faces of the highest dimension $k = m - 1$ and then calculate search vectors for lower k -dimensional faces as arithmetic mean of search vectors from adjoining $(k + 1)$ -dimensional faces. Fig. 4 shows search vectors v for two 2-dimensional faces and two 1-dimensional faces (edges) of our example from Fig. 2.

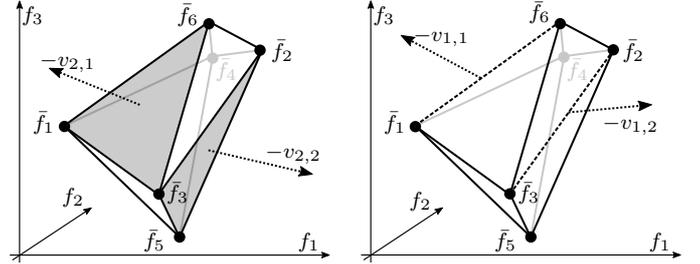


Fig. 4. Search directions v for two faces (left) and two edges (right).

E. Solving Subproblems

Now, for each base point b with associated search direction v , the scalarized problem (GA) has to be solved. Its solution yields a boundary point of the CFPS. Fig. 5 presents the execution of subproblems for the example from Fig. 2. Here, each subproblem is executed sequentially. This procedure can be partially *parallelized* as we will explain in Sec. III-G.

F. Approximation

For the final approximation of the CFPS, we use a Delaunay triangulation—which is actually intended for convex spaces. The Quickhull algorithm [23] provides a well-tested implementation for this, by providing a mapping of point lists for neighborhood relations (Fig. 5). Due to the successive computation of subproblems, it is possible to store point enumerations for each subproblem dimension k and adopt them up to the highest dimension $k = m - 1$. Together with the set of all computed boundary points of \mathcal{P} , a triangulated approximation

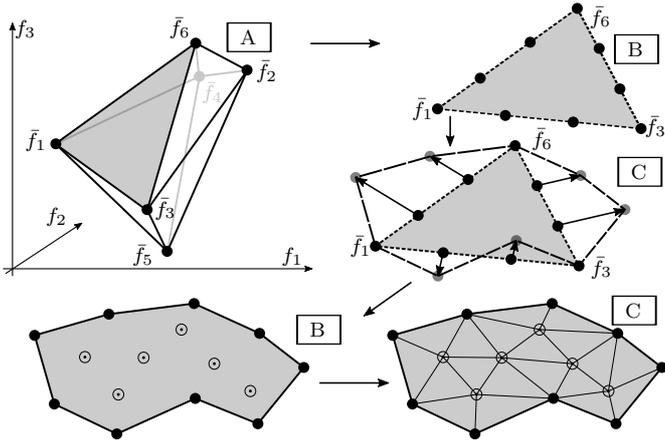


Fig. 5. Solution of subproblems: choose subproblem (A), generate evenly distributed base points (B), solve the related scalar optimization problems (C).

of \mathcal{P} can be created. In Fig. 6, the approximation steps for increasing subproblem dimension $k \in \{0, 1, 2\}$ are shown for a mathematical example from [6, Sec. 9.1]:

$$\begin{aligned} \min_{x_1, x_2, x_3} \quad & F(x_1, x_2, x_3) = [x_1, x_2, x_3]^T \quad (\text{Ex}) \\ \text{s.t.} \quad & x_1 \cdot x_2 \cdot x_3 \geq 300, \quad x_1^2 + x_2^2 + x_3 \geq 119 \\ & 0 \leq x_1, x_2, x_3 \leq 10 \end{aligned}$$

In the presented tool, the number $N \in \mathbb{N}$ of base points at the longest edge is used as a setting option to control the *accuracy* of the computed approximation, i.e., the number of resulting base points n_b and therefore the number of computed boundary points of the CFPS by solving Problem (GA). Table I shows that with increasing N , the total number n_b of computed boundary points increases more than linearly for Example (Ex). The computational time of our presented method grows proportionally with n_b —as expected.

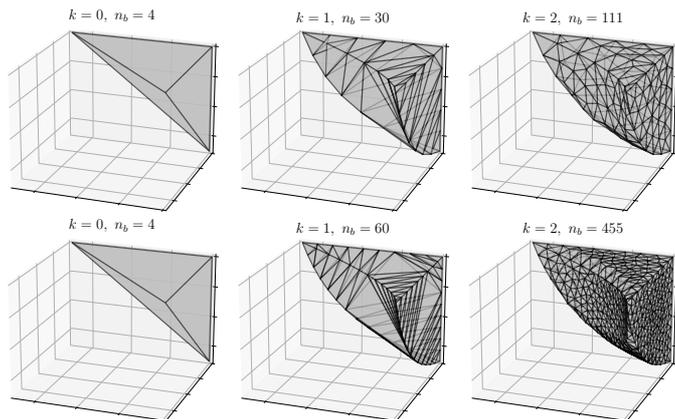


Fig. 6. Successive approximation of the image space of (Ex) with $N = 6$ (top row) and $N = 12$ (bottom row). n_b is the total number of base points b .

G. Parallelization

Once all subproblems for a dimension k are defined, the numerical solution of the associated scalarized optimization prob-

TABLE I
COMPARISON OF DIFFERENT N AND RESULTING NUMBERS n_b FOR (Ex).

N	2	4	6	8	10	12	14	16
n_b	19	59	111	198	305	455	741	1089
CPU-time [s]	6	15	30	70	140	219	407	515

lems (GA) is independent of each other. Only the generation of base points for subproblems of the higher dimension $k + 1$ depends on the computed solutions of dimension k . Therefore, it is appropriate to compute the solutions for all k -dimensional problems *in parallel* (see also [20]). Since the evaluation of the constraints $c(x)$ as well as the objective function $F(x)$ requires computational-time-intensive Spice-simulations, it is also advisable to store all computed points. Dependent on the number of available processes, various parallelization schemes are possible. Thereby, the number of parallelly considered k -dimensional subproblems as well as the number of parallelly solved optimization problems (GA) for each subproblem can be chosen. The efficiency of those variants will be discussed and compared to the original sequential approach in the following section.

IV. EXPERIMENTS

A. Application: Topology Selection

1) *Example Circuits*: We selected a group of similar *operational transconductance amplifiers* (OTA, see Table II) with different topology complexities to show the applicability of the presented CFPS approximation method. All OTAs have a *single amplification stage* (1) with an *NMOS differential pair* (N). We have varied the load type of the differential input pair, which is indicated by capital letters. The different types are a *single current source* (A), a *current mirror* (B), a *cascode current mirror* (C), and a *modified Wilson current mirror* (D). It is expected that these variations will primarily improve the gain or increase the gain-bandwidth with respect to the Wilson current mirror. In addition, a *folded cascode stage* (fc) is added to improve gain, but at the cost of higher active area requirements. For the computation of two-dimensional performance spaces of the selected OTAs, *gain* (A_{V0} in dB) and *gain-bandwidth* (f_{GBW} in Hz) are chosen as performance criteria of interest. For three-dimensional performance spaces, we add the *active area requirement* (area in m^2) as a third criterion.

TABLE II
SELECTION OF DIFFERENT OTA TOPOLOGIES

Key-Name	Features
OTA-A-N-1	Simple current source load (A)
OTA-B-N-1	Current mirror (CM) load (B)
OTA-C-N-1	Cascode CM load (C)
OTA-C-N-1-fc	Folded cascode stage (fc) with CM load
OTA-C-N-1-fc-ci	Folded cascode stage (fc) with CM load and cascode current source (ci) for the diff. input pair
OTA-D-N-1-fc-ci	Folded cascode stage (fc) with CM load, improved Wilson CM and cascode current source (ci)

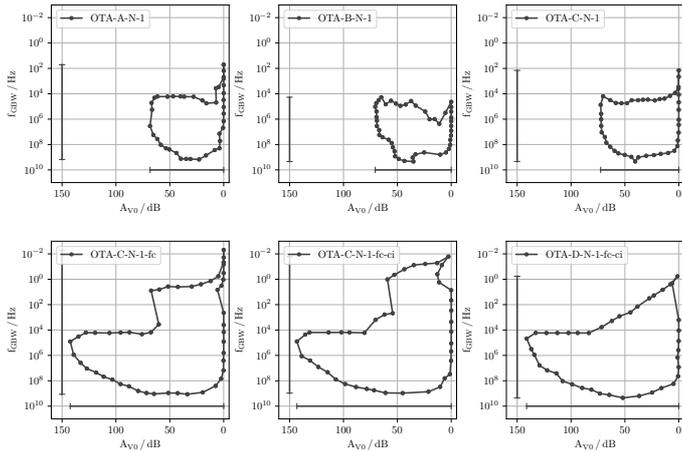


Fig. 7. Two-dimensional performance spaces for OTA topologies with $N = 15$ and an increasing complexity from upper left to bottom right. Load capacitance: $C_L = 1$ pF; $V_{DD} = 3.3$ V; $T = 27^\circ\text{C}$; Technology: 180nm Standard CMOS.

TABLE III
INDIVIDUAL BEST PERFORMANCES FOR EACH OTA.

	A	B	C		D
best value	–	–	–	fc	fc-ci
A_{VO} / dB	68.3	70.8	72.6	142.5	141.9
f_{GBW} / GHz	1.507	2.641	2.155	1.150	0.938
area / μm^2	–	16.0	–	–	59.8
n_b (2D)	42	48	47	50	43
n_b (3D)	–	268	–	–	197

2) *CFPS Computation*: First, we approximated two-dimensional fully realizable performance spaces for all OTAs, where gain competes with gain-bandwidth. Fig. 7 shows the results of our developed tool with $N = 15$. As expected, the CFPS increases with higher complexity of the circuit topology, i.e., the gain increases with the complexity of the current load and the added folded cascode, while there is no loss in gain-bandwidth. A computation of the three-dimensional performance spaces of an OTA with low complexity (OTA-B-N-1) and an OTA with high complexity (OTA-D-N-1-fc-ci) shows that the advantage of higher gain arises at the expense of a larger active area requirement due to the higher number of transistors, see Fig. 8 ($N = 8$). With the help of two-dimensional projections of the three-dimensional CFPS, the higher amount of required active area of the complex folded cascode OTA can be determined. Table III depicts the component-wise best performances for all considered OTAs as well as the total number of computed boundary points n_b for the final approximation of each CFPS. For all computed two- and three-dimensional approximations, a *uniform distribution of boundary points* can be observed.

B. Computational Effort

The computation of the CFPS—and already of its Pareto front only as, e.g., in [6, Sec. 9.2]—is extremely expensive regarding computational time. Even for a low number of performance criteria $m \in \{2, 3\}$ and a coarse approximation quality,

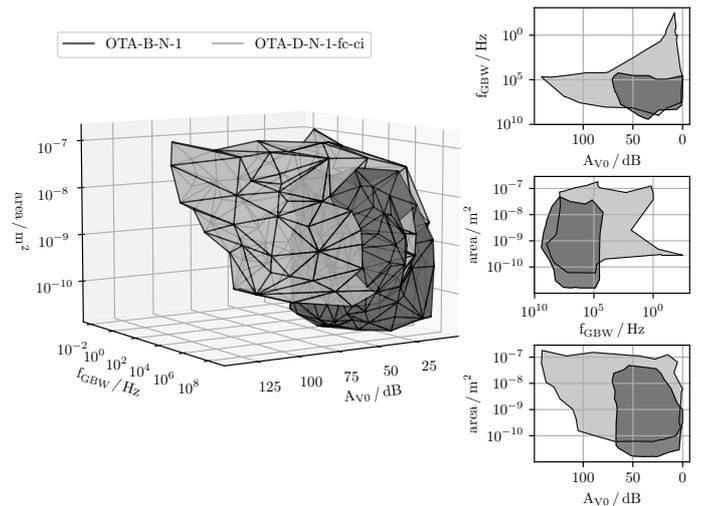


Fig. 8. Three-dimensional performance spaces ($N = 8$) for two OTAs with different topology complexity and their two-dimensional projections.

controlled by N , the computation may take several hours up to a few days (also depending on the circuit complexity), if no parallelization is used. Therefore, we will use the sequential approach (see e.g. [19]) as a baseline and show the efficiency of different parallelization schemes for solving the generated subproblems in the following.

1) *Hardware and Implementation*: For our experiments, we use an Intel[®] Xeon[®] E5-2667 v3 CPU with 2×16 cores and 128 GB of RAM. As SPICE-simulator, Cadence[®] Spectre[®] 18.1.0.077 is used. The numerical solution of the (generally non-convex) optimization problems (1) and (GA) is done by a basin-hopping algorithm, which is a global optimization technique that combines a stochastic global stepping approach with a local minimization method (SLSQP). The implementation of the developed and presented tool is mostly done in Python 3.8.

2) *Results*: The computational effort is twofold. On the one hand, depending on the choice of N (see Sec. III-F), the number n_b of base points b and, therefore, the number of optimization problems (GA) for computing boundary points of the CFPS grows (see also Table I). On the other hand, for the numerical solution of each problem (GA), several SPICE-simulations have to be done. Actually, for each iteration of the local optimizer, two simulations are necessary: one for evaluating the objective function and one for evaluating the constraints. Those SPICE-simulations are time-consuming and become more expensive with increasing complexity of the underlying circuit.

For OTA-B-N-1 (low complexity) and OTA-D-N-1-fc-ci (high complexity), we compute the three-dimensional CFPS (gain vs. gain-bandwidth vs. area requirement) for $N \in \{4, 8\}$. In Table IV, we compare the sequential computation with two parallelization schemes. The first scheme (1, 8) considers *one subproblem* after the other, while solving *eight base points* in parallel. The second scheme (8, 8) considers *eight subproblems* in parallel and solves *eight base points* for each subproblem in parallel—having a total of 64 processes in parallel. The greatest

TABLE IV
COMPUTING TIMES [HH : MM] FOR DIFFERENT PARALLELIZATION SCHEMES.

N	scheme	OTA-B-N-1		OTA-D-N-1-fc-ci	
		time	n_b	time	n_b
4	(1, 1)	15:31	53	115:15	48
	(1, 8)	13:41	70	58:07	43
	(8, 8)	7:11	69	14:20	64
8	(1, 1)	— ⁽¹⁾	—	— ⁽¹⁾	—
	(1, 8)	50:40	348	160:40	453
	(8, 8)	25:03	464	23:24	286

(□, □) (parallel subproblems, parallel base points per subproblem)
—⁽¹⁾ computation not possible within 12 days

time gain can be achieved by the full parallelization scheme—as expected. Compared to the sequential approach, the (8, 8)-scheme is about *twice as fast* for OTA-B-N-1 and even about *eight times as fast* for OTA-D-N-1-fc-ci in the low accuracy setting $N = 4$. The results are even more impressive in the high accuracy setting $N = 8$, since without parallelization, the computation for neither of both OTAs was possible within twelve days. With full parallelization, we obtained a fine approximation for each OTA in about one day. Overall, the time gain through (full) parallelization was larger for approximating the CFPS of OTA-D-N-1-fc-ci. Here, due to its high complexity, SPICE-simulations are more expensive, while the effort for the remaining optimization process keeps similar.

Due to the fact that our modified CVT-method (see Sec. III-C) contains stochastic elements, the number of base points distributed on the surfaces of the CFPS-approximation ($k = 2$) varies in each computation. Therefore, we also included the total number n_b of computed boundary points of the CFPS in Table IV, since this also influences the computational time.

V. CONCLUSION

We have presented a general approach for approximating the complete feasible performance space (CFPS) of integrated analog circuits, which bases on the well-established normal boundary intersection method. Thereby, we propose models for a reliable comparison of analog circuit topologies. Designers of mixed-signal systems can benefit from graphical information on all attainable performance parameters. Parallel to this work, an interactive tool with out-of-the-box functionality has been developed for the computation and exploration of performance spaces. It can be connected to any numerical optimization software as well as any SPICE-simulator. The efficiency of different implemented parallelization schemes has been shown for real use cases. Due to parallelization, performance spaces up to dimension three can be computed with high accuracy within in a reasonable time. Such computations would not be possible by applying the conventional, sequential approach. For future works, our parallelized method also allows the computation of higher dimensional performance spaces, where more than three performance criteria compete.

REFERENCES

- [1] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 640–669, 2007.
- [2] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, and S. Zizala, "WiCkeD: analog circuit synthesis incorporating mismatch," in *Proceedings of the IEEE 2000 Custom Integrated Circuits Conference*, pp. 511–514, 2000.
- [3] V. Pareto, *Manuale di Economia Politica*. Società Editrice Libreria, 1906.
- [4] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [5] G. Stehr, H. Graeb, and K. Antreich, "Performance trade-off analysis of analog circuits by normal-boundary intersection," in *Proceedings 2003. Design Automation Conference*, pp. 958–963, 2003.
- [6] D. Mueller-Gritschneider, H. Graeb, and U. Schlichtmann, "A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 915–934, 2009.
- [7] R. S. Motta, S. M. Afonso, and P. R. Lyra, "A Modified NBI and NC Method for the Solution of N-Multiobjective Optimization Problems," *Struct. Multidiscip. Optim.*, vol. 46, no. 2, pp. 239–259, 2012.
- [8] S. Siddiqui, S. Azarm, and S. A. Gabriel, "On Improving Normal Boundary Intersection Method for Generation of Pareto Frontier," *Struct. Multidiscip. Optim.*, vol. 46, no. 6, pp. 839–852, 2012.
- [9] K. Khaleedian and M. Soleimani-damaneh, "A new approach to approximate the bounded pareto front," *Mathematical Methods of Operations Research*, vol. 82, no. 2, pp. 211–228, 2015.
- [10] M. Krasnicki, R. Phelps, R. Rutenbar, and L. Carley, "MAELSTROM: efficient simulation-based synthesis for custom analog cells," in *Proceedings 1999 Design Automation Conference*, pp. 945–950, 1999.
- [11] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog Circuit Design Optimization through the Particle Swarm Optimization Technique," *Analog Integr. Circuits Signal Process.*, vol. 63, no. 1, pp. 71–82, 2010.
- [12] R. Vural and T. Yildirim, "Analog circuit sizing via swarm intelligence," *AEU - International Journal of Electronics and Communications*, vol. 66, no. 9, pp. 732–740, 2012.
- [13] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137–148, 2009.
- [14] M. Barros, J. Guilherme, and N. Horta, *Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques*. Springer Publishing Company, Incorporated, 1st ed., 2010.
- [15] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An Efficient Bayesian Optimization Approach for Automated Optimization of Analog Circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2018.
- [16] D. Schreiber, B. Ohse, J. Kampe, and C. Schneider, "Performance Space Supported Design of Analog Electronic Circuits," in *Proceedings of 10th Vienna International Conference on Mathematical Modelling*, 2022.
- [17] D. Schreiber and J. Kampe, "OTA Performance Space Modeling for System-Level Optimization of SC-circuits," in *2020 27th IEEE International Conference on Electronics, Circuits and Systems*, pp. 1–4, 2020.
- [18] T. Massier, H. Graeb, and U. Schlichtmann, "The Sizing Rules Method for CMOS and Bipolar Analog Integrated Circuit Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 12, pp. 2209–2222, 2008.
- [19] D. Schreiber and J. Kampe, "On Applying Pareto Optimization for Complete Performance Space Modeling of Analog ICs," in *ANALOG 2018; 16th GMMITG-Symposium*, pp. 1–6, 2018.
- [20] D. Mueller, H. Graeb, and U. Schlichtmann, "Trade-off Design of Analog Circuits Using Goal Attainment and "Wave Front" Sequential Quadratic Programming," in *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '07*, pp. 75–80, EDA Consortium, 2007.
- [21] F. Gembicki and Y. Haimes, "Approach to performance and sensitivity multiobjective optimization: The goal attainment method," *IEEE Transactions on Automatic Control*, vol. 20, no. 6, pp. 769–771, 1975.
- [22] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.
- [23] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.