CAD: AIDA-TOP - Analog IC Topology Selection

João Melo, Ricardo Póvoa, and Nuno Horta, *Senior Member, IEEE*

***Abstract*— This Master thesis describes the development of a technology independent, highly customizable topology selection methodology compatible with AIDA-C. Topology synthesis is an essential step in circuit design, majorly dependent on the knowledge of experienced designers. Producing an EDA tool in this task of a circuit's design flow would drastically improve its time-to-market, since many other elements in this flow are already automated. The method chosen uses the information of past sizing optimizations, combining them with MCDM algorithms to get the most apt topology. It does so if there exists an optimized circuit able to reach the desired performances, or otherwise.**

The topology selection method was implemented as the AIDA-TOP program. It was tested using AIDA-C optimizations of four distinct topologies, belonging to the CMOS OTA family. The program was verified to progress as expected with diverse input examples. The tool used 40 test optimizations with multiple simulated circuits each to arrive at an overall accuracy rate of 76,41%.

Keywords— Analog circuit design; Topology Synthesis; Electronic Design Automation; Topology Selection; Multi-objective optimization; Multi-criteria decision analysis.

I. Introduction

With the invention of the transistor, and the ability to integrate it in large numbers into a semiconductor made integrated chip (IC) electronic circuits adoption became mainstream. To understand the magnitude of their expansion, the semiconductor market worth grew from $20 billion to $480 billion [1], [2] (approximately), from 1986 to 2018.

Despite the relatively small proportion of analog circuits in electronic devices, their design is more resource intensive than the digital counterpart. This complexity stems intricate relations between devices, and performance goals and specifications being continuous.[3] The present difficulty in the development of Analog and Mixed-Signal (AMS) circuits, is a threat to product life cycles due to the increasing time-to-market-constraints. Because of the continuous nature of the "real world", AMS circuits emerge as a necessity to interface with digital systems. This connection links the successes of IC and semiconductor market, with the ability to produce this category of circuits.

In comparison to other device types within the semiconductor industry, analog ones are forecasted to have second largest rate of growth until 2023 (7.4%) [4]. The future interest in this type of circuits, and existing limitations to their conception, justify the importance of improving AMS IC design.

The set of actions that start with the concept for a circuit and end with the delivery of a physical circuit is its design flow. [5] When one examines the topological side of one of the design stages named "Cell design" one confronts a task that still relies on graphical schematic entry tools, a manual chore relying on the expertise and time of highly trained analog designers. In view of this handicap, the current work proposes a methodology to improve it.

Upon this task's conclusion, the range of achievable performance specifications is much narrower. For this reason, it is paramount, that an adequate topology is chosen, as a failure to do so limits the potential of a circuit.

A wealth of ways to approach this problem have been attempted, some deficient due to excessive computational requirements, others due to the restrictive library, or due to the time necessary to compose a model.

To fill the breach in contemporary IC design, it is offered a methodology enabling the creation of a knowledge database (library), that can be customized with the most useful solutions for multiple sectors and uses. Multi Criteria Decision Makers (MCDM) are adopted, summarizing the optimizations, into a smaller set of points that can describe them and be used to choose topologies competently. The methodology is implemented into a tool named AIDA-TOP which is also used to test the methodology.

This document is organized as follows. To start, Section II has a literature review. On Section III the architecture of the methodology is approached. Section IV has the preparations done to validate the methodology through the creation of a tool. Section V shows results achieved from using said tool. Finally, in Section VI the conclusions are addressed.

II. Previous Works

A. Introduction

In this section works that introduced paradigms that were important for solving the problem of topology creation are seen chronologically, describing the evolution set forward by them.

B. Literature Review

Typically works can be distinguished into two disjunct categories: The process of selecting the most appropriate topology amongst known ones, is named **Topology Selection**; the other way is to construct one or more topologies, combining components or blocks of components, called **Topology Generation**.

In the first notable work, multiple different schematics were available to be picked by the user. Consequently, the process of selection can be labeled as **manual**. IDAC [6] performed sizing and topology selection separately, so it can be categorized as **independent** topology design. It saved time by having an incorporated library with useful solutions, ready to be used. However, in this first approach, the selection relied on the user to choose the relevant topologies and interpret the results of the sizing tool.

Harjani et al. [7] developed a new system, OASYS, that subdivided a cell into reusable sub-cell abstracts blocks (e.g. current mirror), sub-blocks originating the **hierarchical** perspective. This permitted the usage of the same blocks in different contexts, setting them up only once. This representation did remove the ability to employ design "tricks", that affect multiple blocks, which are only accessible when single devices are exposed and independently mutable.

This tool applied methods that reduced computation, while still providing the best or at least a good option. The authors instilled subject specific information, to be used for synthesis. This information had to be manually modeled and introduced into the tool, making it a **manually compounded knowledge-based** tool.

Contrary to IDAC, in OASYS its design flow did not demarcate the topology design part from other tasks, being a **joint** topology creation tool.

In OPASYN [8] after selection, the topology was forwarded to the sizing module where the parameters were tuned, maximizing a design cost function, calculated as a weighted sum of the performance in each parameter. This method of evaluating the success of a circuit departs from the previous works that focused solely in attaining the constraints presented. It was an **overall** performance objective, contrary to the previously seen focus on **constraints achievement**.

Maulik et al. [9], realized Cell design could be posed Nonlinear Programming problem. KVL and KCL equations to be solved were extracted resorting to an external algorithm, initiating the notion of **automatic compounded knowledge-based** tools, even if used only for a portion of knowledge.

The objective function used here was centered on minimizing the area of the circuit, the **fixed single** objective of this optimization solution, continuing the search for designs that perform better in this sole objective even after ensuring the minimum acceptable area. The set of all solutions where the entirety of constraints is guaranteed is the feasible region (with feasible solutions in it), within which an optimum is sought.

Lohn and Colombano [10] opted to use an Evolutionary **Genetic Algorithm** (GA) for the task of defining a topology. A GA belongs to the field of evolutionary computation which bases itself on the Darwinian concept of survival of the fittest. In this version, the individuals that integrated the population were the circuits.

The initial creation of the population, as well as subsequent generations, have the potential to originate unseen circuits, revealing the first **topology generation** tool. This work gave the user the ability to select the specification considered a priority thus having a **mutable single** objective.

Until this work, the way to evaluate the performance of a circuit, had been through numerical analysis of the circuits. The analysis was performed on circuit models, requiring previous set-up of equations that described them. This technique is **equation-based** evaluation. Lohn and Columbano [10], however, used the Simulation Program with Integrated Circuit Emphasis (SPICE) tool for accurate electrical simulation. It is thus possible to categorize the evaluation of circuits produced as **simulation-based**. This latter method is more accurate yet more computationally demanding.

Sripramong et al. [11] pointed out shortcomings displayed by the previously overviewed GA, which it vowed to improve using **Genetic Programming** (GP). Its schematics were represented in tree-form, conforming to the standard GP representation.

A **multi-objective evolutionary algorithm** (MOEA) joins Multi-Objective perspective and an EA one first exemplified by MOJITO [12], which was published in 2011 and uses the **NSGA-II** algorithm [13].

A **multi-objective optimization** (MOO) problem is one that, contains more than one objective function. The concept of feasible region is also applicable in MOO. The challenge of finding an optimal analog circuit, can be seen as such, since designers usually have multiple criteria they wish to optimize e.g. minimize area and maximize gain.

One concept unique to MOO is a MCDM (multi-criteria decision maker), Pareto-Optimality. Choosing any two solutions, it can be noted that one is superior (dominating) to the other in all objectives (dominated). Solutions are non-dominated when they are not worse than any solution in all objectives. The curve obtained from line-connecting these solutions, is the pareto-optimal front (PoF) and the group of all solutions the Pareto Set. In Fig. 1. there is an example of a pareto front and dominated points that are excluded from it. In this case the axes evolve in decreasing preference, so the preferred points are the minima.

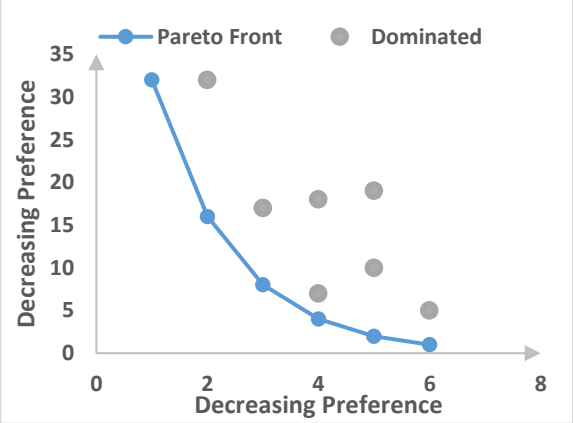

Fig. 1. Pareto Front Example.

Li et al. presented in 2018 InnovA [14], a cognitive architecture applied to analog circuit design. Artificial cognitive systems look to incorporate cognitive neuroscience and developmental psychology to replace the manual or algorithmic methods of knowledge acquisition and once used.

This work, tries to decompose circuits, understand the cause for the existence of its parts, and integrate these skills with the ability of learning to learn. The use of cause and effect for synthesis and learning introduces **causal** synthesis.

This recent tool resorts firstly to selection of topologies, generating multiple circuits only upon failure to encounter a job fulfilling circuit. For its use of both strategies it can be categorized as a **hybrid** solution regarding the discovery process.

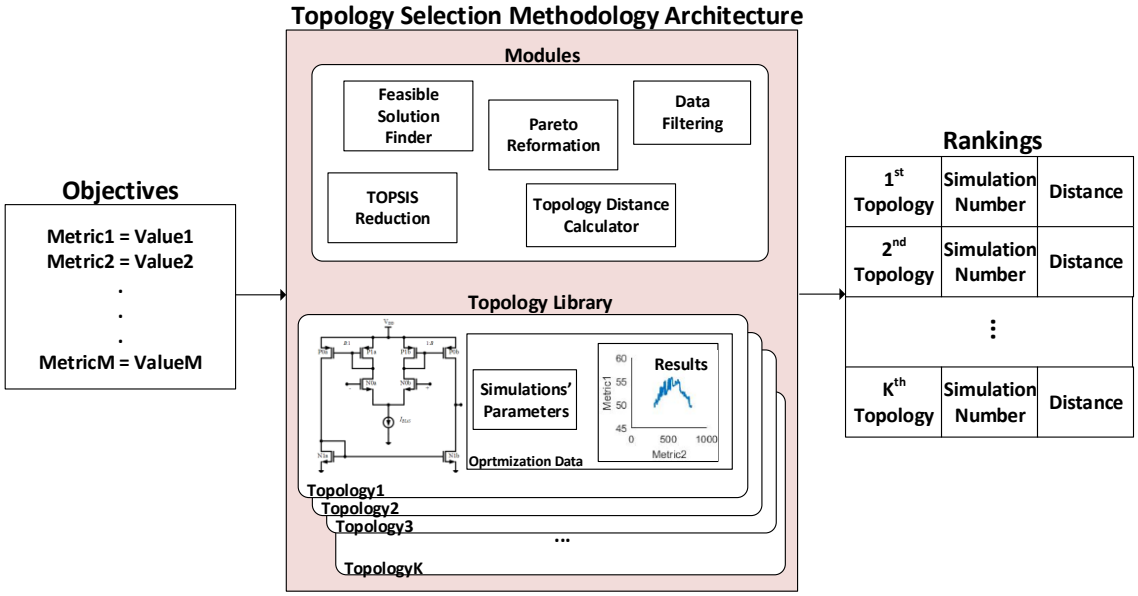

Fig. 2 Topology Selection Methodology Architecture.

C. Categorization of solutions

The categories presented throughout the historical perspective were used to classify notable works. Additionally, two quantitative metrics are included, to give a perception of the variety of topologies and real-world usage of the tools.

D. Work Proposal

It is proposed a topology selection methodology, with flat hierarchy, using previous AIDA-C optimizations to improve multiple objectives simultaneously, and its execution independent from other design flow stages. A tool is developed to be used to complement AIDA-C, which implements the methodology developed.

Concerning the main algorithm employed, it can be defined as knowledge-based, relying on stored optimizations to, from the results of their objectives, supply which topologies are more appropriate to perform as requested. This approach allows for the tool to provide a lot of flexibility, permitting each user to store and search only optimizations of circuits, technologies and measurements that it considers valuable. Furthermore, there is no need for intricate models that require a large time investment to incorporate, nor does it need long execution times to select a topology since the data it requires was previously saved and the algorithms used are simple.

III. System Architecture

A. Introduction

A methodology was created to recommend the most likely, from a set of topologies, to succeed in behaving within the desired parameters.

To arrive at this ranking of topologies, a series of modules were designed. A set of AIDA-C optimizations with the netlists and technology files it used, also have to be available (library), and the user must provide which are the relevant metrics and respective values desired (objectives). This architecture can be seen in Fig. 2.

The user inputs and library data go through the modules in the sequence detailed in Fig. 3. to deliver the output rankings. This sequence starts by firstly filtering the optimizations' data in the library to remove parameters not regarded as objectives and circuits that do not contain data for all the objective metrics (Data Filtering module). Then, it encounters the Feasible Solution Finder, where there is a search to get points complying with objectives' specifications. If solutions are found, then only those are considered in the remaining modules, otherwise all library solutions are used.

The program resumes, moving to the Pareto Reformation module, getting the topologies optimizations' results to pareto front form. From there, each topology it is found its extreme and compromise points through the application of Technique For Order of Preference by Similarity to Ideal Solution (TOPSIS). There are then two Topology Distance Calculators. The Closest and the Farthest Topology Distance Calculator.

The first is used when there are no feasible solutions, the Farthest if there are any. In the Closest Distance Calculator, line segments between the TOPSIS module points are created. The distance from the goal to all line segments is computed. Each topology's closest point, from those that that existed in the reformed pareto, is kept, associating to it the line segment distance value. Then the points are sorted from closest to farthest. In the latter calculator the process of adding line segments is not done, only computing and storing every topology's farthest point sorted by decreasing distance.

The reason for having opposite criteria in these calculators is that if the search for feasible solutions was successful, the goal is to deliver topologies sorted by their ease in reaching the objectives, represented by larger distances to them. The topology pertaining to the farthest point from the objective is output first, and for all elements the respective distance and simulation number are provided as well. On the other occasion, the aim is to suggest, from the library, the list of topologies starting with most suited to be optimized or tweaked to fulfill all goals, ending with the least likely. The closest topology is then hypothesized as most suitable, because it will require less improvement. In this case the order is reversed, also indicating the configuration that is more prone to achieve the intended results.

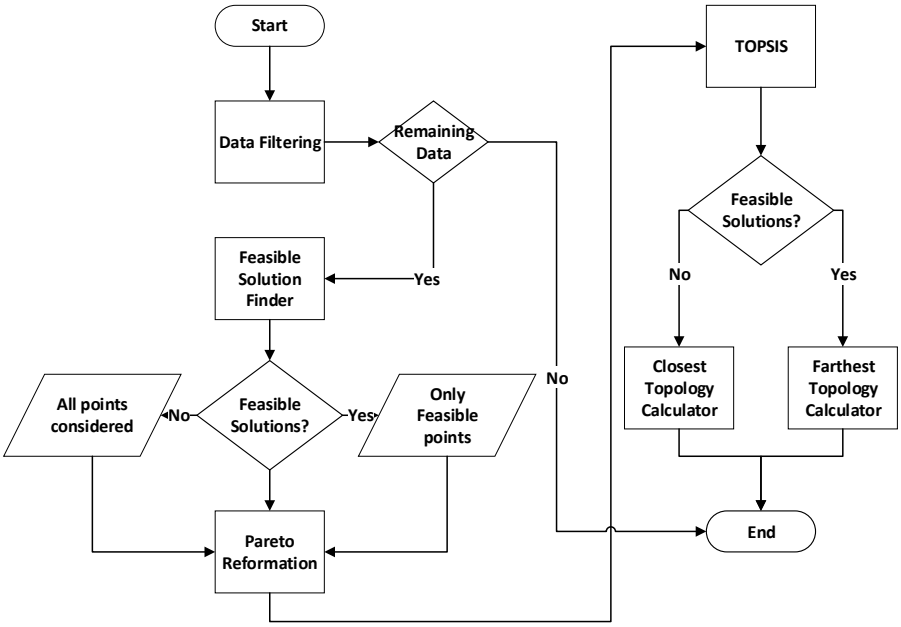

Fig. 3. Module sequence in Topology Selection Methodology.

B. Library

Two components are required in a library element:

- Definition of a topology.
- Results from the topology's optimizations.

The topological definition must contain information of what devices are used, their connections, and their technology. When it comes to the results from the optimizations, they are made up of several simulations. For each of them, the performance of chosen objectives is recorded together with the sizing of electrical components that led to them.

The sizing and topological definition are necessary for recreating, as well as altering the selected solutions, making it possible to take advantage of the output provided.

C. Modules

1) Relevant Data Filter

Should there be any optimizations in the library that contain all goal parameters asked, then the execution continues, excluding the topologies that do not, and keeping the ones that do, limiting them to have only the input parameters. Otherwise, there is no useful information in the library and the methodology terminates.

2) Feasible Solution Finder

This part of the method goes through all simulations within the optimizations, comparing the simulation values to the objectives'. If a simulation is better in all the objective metrics it is deemed a feasible solution and it is stored, if it is not, it is discarded.

3) Pareto Reformation

The removal of irrelevant measurements that took place previously, may have left some dominated points. In this module, the data is restructured so that each topology contains only a pareto set. To identify from the non-dominated points, Algorithm 1 is used. It obtains non-dominated set P' from set P, which has N elements. This algorithm considers all elements of P as non-dominated at the start, copying them to the non-dominated set P'. It then iterates through all the points i in P. If P_i is still present in the non-dominated set P', it must be compared with all the points that succeed it in P (represented by j), to confirm its non-domination. In this comparison, should either of the points reveal itself as dominated (i or j), then it is removed from P', continuing to the next element of P. In the end P' contains a POF. This algorithm is very similar to the second approach shown in the chapter from the book by Deb et al [15].

4) TOPSIS

TOPSIS is a MCDM method by Yoon and Hwang[16]. It uses as criteria the distance to an ideal solution to find a compromise point. An ideal point is one that combines the best performing elements in each of the existing dimensions. An ideal point can be defined as:

$$A^* = \left\{ \left(\max_i x_{ij} \,\middle|\, j \in J^* \right), \left(\min_i x_{ij} \,\middle|\, j \in J^- \right) \mid i = 1,2,\dots N \right\} \quad (1)$$

i as the solution x number from N available. Opposite to this the negative ideal point is:

$$A^- = \left\{ \left(\min_i x_{ij} \,\middle|\, j \in J^* \right), \left(\max_i x_{ij} \,\middle|\, j \in J^- \right) \mid i = 1,2,\dots N \right\} \quad (2)$$

with the same variables as before. In these points from M total criteria there is:

$$J^* = \{j = 1,2,\dots s \mid j \text{ represents benefits criteria}\}$$

$$J^- = \{j = 1,2,\dots t \mid j \text{ represents costs criteria}\}$$

Algorithm 1 –Pareto Reformation Algorithm.

$P' = P$, Copy all solution set P to set P'

$i = 1$

Until $i \leq N$, While there are solution points to be evaluated

- **If $P_i \in P'$,** If solution i of P still exists in the P' set
  - **$j = i + 1$,** Set solution iterator j for the point following i in set P
  - **Until $j \leq N$,** While solution hasn't been compared to all following ones in P
    - **If P_i dominates P_j**, P_j worse in all components
      - **$P' = P' \backslash \{P_j\}$**, Remove j solution from P'
      - **$j = j + 1$,** Compare P_i with solution following P_j
    - **Else if P_j dominates P_i,** P_i worse in all components
      - **$P' = P' \backslash \{P_i\}$,** Remove solution i from P'
      - **$i = i + 1$,** Check next solution of P is nondominated
      - **$j = N + 1$,** No need to compare with further elements since it is removed
- **Else,** P_i had been excluded from P' for being dominated
  - **$i = i + 1$**,

P' now only has Non-dominated solutions

In this algorithm the choice is made considering the distance from each solution to both these points, using closeness metric:

$$C_i = \frac{d_{iA^-}}{d_{iA^-} + d_{iA^*}}, i = 1,2, \dots, N \qquad (3)$$

Where d stands for the Euclidean distance. The point with largest closeness factor is the compromise point.

Due to the potential discrepancy in orders of magnitude on the criterions used, the distances require normalization. Normalization is applied to the initial solutions p_i from pareto P by dividing all elements' components j by the respective sum square roots across all p_i solutions.

$$x_{ij} = \frac{p_{ij}}{\sqrt{\sum_{i=1}^{N} p_{ij}}} \qquad (4)$$

The solution with maximum closeness (compromise) is saved, additionally storing the extremes, which are the points who best perform in a single objective.

The steps to arrive at these ultimate sets are described in Algorithm 2 creating the TOPSIS algorithm. This is applied to all K topologies from the ones left. To start, the points that have the best performance in the jth objective ($j = 1,2, \dots M$) are saved into reduced set R, at the same time calculating the jth coordinate of ideal and anti-ideal points A_j^* / A_j^- , and this coordinate's normalization factor $normFact_j$. Upon conclusion, this normalization factor is applied iteratively to jth coordinate of the anti-ideal and ideal-points, and to all N pareto points p_i . Having all normalized elements, the distances from ith point to ideal and anti-ideal points are calculated, as is the respective closeness factor. Finally, the point with the compromise C_i, maximum closeness, is saved into reduced set R . The complexity per topology of this algorithm for the is $O(MN)$.

5) Closest Topology Calculator

This module is used for the getting the closest topology in the event of not having found any feasible solutions. To obtain the preferred solutions, a metric based on the Euclidean distance to line segments (created by connecting the dimensional optima to the compromise point), is used. The line segments represent a linear approximation of the performance that a topology can realize.

If the best point was not in the reformed pareto (P'), it is an added point. In such case, the closest pre-existing point must be indicated as a solution. These additional points only serve to supply a more useful distance metric, not having associated any actual replicable device sizing. Due to this, the distance to this added point is registered (if best), as the distance to the topology itself, yet the simulation that the program outputs has to contain a specification, allowing the user to work from it. Therefore, the distance value to all topologies is sorted using both added points and simulation-based ones (P' points), but once the topology ordering is achieved, the topologies' nearest simulated point is singled out, to be displayed in the output.

The distance used was said to be Euclidian distance based, but it requires a modification. Since this module mirrors the improvements needed so that all objectives are compliant, if in some of the metrics the performance is already adequate, no effort is required in those metrics. So, the modification is that only the distances of the non-attaining objectives matter, contributing to the overall distance, since the other components are already ensured. It is the one with smaller cumulative distance in normalized non-conforming specifications that needs to be improved less than the others.

This module provides an inexpensive linear approximation of the circuits' performance and then finally calculates and sorts the best topologies, that will be output to the user.

6) Farthest Topology Calculator

When one stands before a feasible solution, the one that is overall farthest is said best. It is assumed to allow more changes in all specifications than the rest of the points.

The distance used is also the Euclidean. Nevertheless, in this module no points are added, instead directly calculating the distance from the target point to the reduced pareto resultant from TOPSIS. After obtaining the distances to these points, feasible solutions are sorted by descending distance, opposite to the other calculator.

Algorithm 2 – TOPSIS Algorithm.

Until $j \leq M$, Until there are no more dimensions

- $R = R \cup \{arg\ \underset{i}{best}\ p_{ij}\}$, Add dimensional bests to reduced set
- $A_j^* = \left(\underset{i}{max}\ p_{ij} \middle| j \in J^*\right), \left(\underset{i}{min}\ p_{ij} \middle| j \in J^-\right)$, Ideal point
- $A_j^- = \left(\underset{i}{min}\ x_{ij} \middle| j \in J^*\right), \left(\underset{i}{max}\ x_{ij} \middle| j \in J^-\right)$, Negative ideal point
- $normFact_j = \texttt{root\_square\_sum}(p_{ij})$, normalization factor
- $j = j + 1$, Next dimension

Until $j \leq M$, Until there are no more dimensions

- $A_j^* = A_j^* / normFact_j$, Normalization of ideal point
- $A_j^- = A_j^- / normFact_j$, Normalization of negative ideal point
- $j = j + 1$, Next dimension
- **Until** $i \leq N$, Until all points have been considered
  - $x_{ij} = p_{ij} / normFact_j$, normalization of paretos points

Until $i \leq N$, Until all points have been considered

- $d_{iA^*} = \texttt{euclidian\_distance}(x_i, A^*)$, Calculate Euclidean distances d_{iA^*}
- $d_{iA^-} = \texttt{euclidian\_distance}(x_i, A^-)$, Calculate Euclidean distances d_{iA^-}
- $C_i = \texttt{closeness}(A^*, A^-)$, Get the closeness C_i
- $i = i + 1$, Next point

$R \cup \{arg\ \underset{i}{max}\ C_i\}$, Add point with maximum C_i to R

IV. Test Library

A. Introduction

A set of circuits, constraints, and objective specifications needed to be chosen, to optimize and get data for assessing the method's success. In this section it is overviewed the elements chosen. Then the implementation of the library follows, consisting on how its elements were encoded. Then, the optimization process and output format is explained. The optimizations performed are described, stating the motivations behind them and the resulting data. After this, all that was required to run the implementation of the method and its validation program was ready.

The family of circuits chosen were OTAs, whose configuration largely resembles the one in OpAmps, the latter being a popular class, deeply linked with the appearance of ICs. OTA's are often less intricate, simplifying the simulation process. [17] The OTAs were implemented with CMOS technology.

Related to the choice of circuits, is the choice of what measures to use. Metrics should not be too similar in all topologies and should also be representative parameters that are used frequently. Furthermore, parameters must not all be simultaneously improved, rather expressing of the trade-offs for which MOO was created for. [18]

B. Measures

For testing, 4 metrics were chosen: Voltage Gain, Figure-Of-Merit (FOM), Offset Voltage (VOS) and Output Swing Voltage (OS). These metrics are briefly explained in the following paragraphs.

1) Voltage Gain

The voltage gain is defined as the ratio between the input voltage v_i and the output voltage v_o which will be delivered to the load.

$$A_v = \frac{v_o}{v_i} [V/V\ or\ dB] \tag{5}$$

2) Figure-Of-Merit

The Figure-Of-Merit term is used as a number that characterizes the performance of circuits in the context of the energy-efficiency and is commonly used in the literature of this sort of topologies.

$$GBW = \{f \in \mathbb{R} | A(f) = 1\ dB\}\ [Hz] \tag{6}$$

$$FOM = \frac{GBW \times C_l}{I_{DD}} \left[\frac{MHz \times pF}{mA}\right] \tag{7}$$

3) Offset Voltage

The offset voltage V_{OS}, is defined, in this work, as the difference between the actual DC voltage that is applied at the output by the amplifier, and the value that would be achieved in an ideal situation (ideal amplifier), which would be half of the positive supply voltage V_{DD}.

$$V_{OS} = V_o - \frac{V_{DD}}{2}\ [V] \tag{8}$$

4) Output Swing Voltage

The output swing voltage is defined as the maximum swing of the output node without generating a defined amount of harmonic distortion. In practice, the OS is determined by the difference between the positive supply and the negative supply voltages or ground minus the overdrive voltages of the transistors that drive the output node, i.e., transistors that form the output branches of the amplifier.

$$OS = 2 \times \left(V_o - \sum_{i=1}^{n} V_{DSAT}^i\right)[V] \tag{9}$$

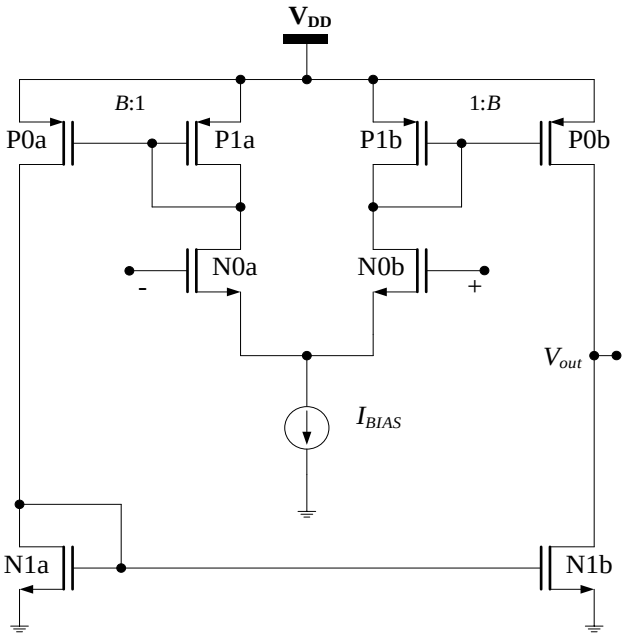

Fig. 4. Symmetrical Amplifier

C. Circuits

For the purposes of this work, four topologies are considered for proof-of-concept: the Symmetrical CMOS OTA, the Telescopic-Cascode OTA, the Folded-Cascode OTA and the Mirrored-Cascode OTA.

The Symmetrical CMOS OTA, shown in Fig. 4., is the most suitable circuit if a high output swing is desired.

The Telescopic-Cascode amplifier (TCA), shown in Fig. 5., often has a gain higher than the Symmetrical CMOS OTA and provides, in general, a good tradeoff between gain, power consumption and speed, but the output swing of this architecture is limited.

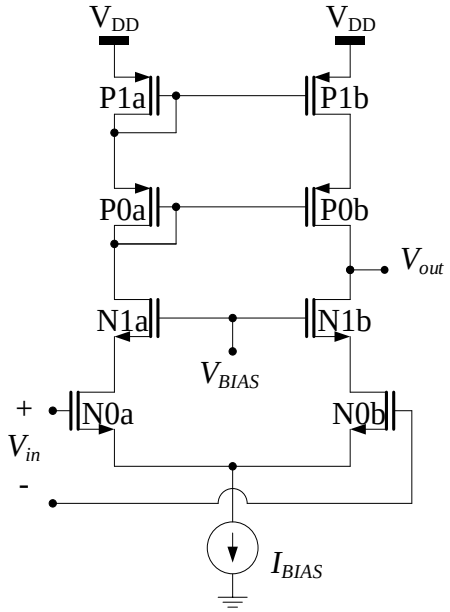

Fig. 5. Telescopic-Cascode Amplifier

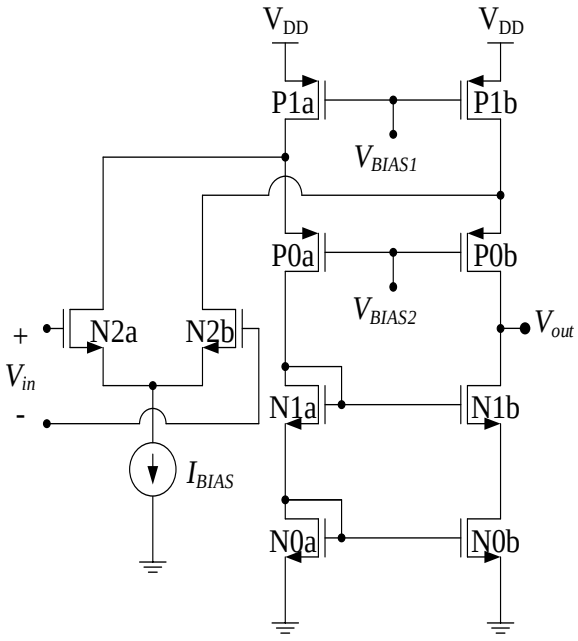

Fig. 6. Mirrored-Cascode Amplifier.

The mirrored cascode amplifier (MCA), shown in Fig. 6. improves gain and the output swing of the amplifier is also improved when compared to the telescopic approach. The major drawback of this topology, when compared to the Telescopic-Cascode amplifier is the power consumption which is higher.

The Folded-Cascode ,Fig. 7., can provide greater OS than the Telescopic-Cascode and increases the common-node input range. However, GBW can be lower.

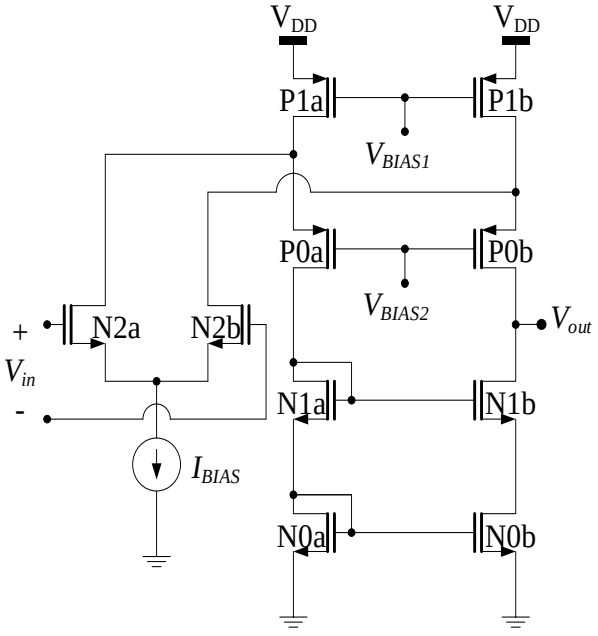

Fig. 7. Folded-Cascode Amplifier

D. Library Implementation

The execution of AIDA-C outputs an XML file, containing a pareto of circuit solutions. Each solution is made-up by the objective metrics' simulation values and the dimensions of transistors the simulation based itself to obtain them. These dimensions are named in this file according to the netlist of the optimized circuit, thus requiring it for re-creating a fully sized version of the circuit. Furthermore, the netlist of the setup for electrical testing (test bench), must also be provided, so that it is possible to know how the performance metrics were measured. All netlists were created to be used with the Eldo® simulator employed within the AIDA-C program, to assess circuit performance. There are also library technology files, which hold the transistors' mathematical model for accurate transistor behavior simulation. The AIDA-C XML files, Eldo® compatible circuit and test bench netlists, and technology files make-up each library item.

E. Optimizations

1) Search Space and Constraints

The search space, since AIDA-C is a sizing tool, has as variables the measurements of the transistors that integrate a circuit

When it comes to constraints there were also set minimum values for important specifications to which these circuits generally must assure: (1) Power Consumption, (2) VOS, (3) Gain, (4) GBW, (5) Phase Margin, (6) FOM. These constraints were set to the same value in all design files, varying only which of them were defined as objectives.

Finally, there are two biasing voltages that each transistor must comply with: delta voltages and overdrive voltages, to maintain the transistors in the desired operating region (saturation).

2) Library Paretos

Within AIDA-C program the objectives to optimize were set to cover all the metric options presented. selecting the population size (128 individuals) and performing four 1000 generation iterations making for a total of 4 thousand generations per circuit. The constraints and genetic algorithm configurations conform to those in state-of-the-art optimizations for the same technology and topologies. The option to arrest optimization at 4000 generation was based on three factors: (1) design experience, (2) by visually observing the progression of an AIDA-C plot indicating the algorithm's progress, (3) the stability of the paretos also plotted in AIDA-C optimizations. [19][20][21]

V. Tests and Results

The modules, their sequence for the topology selection and for testing were programed into MATLAB®.

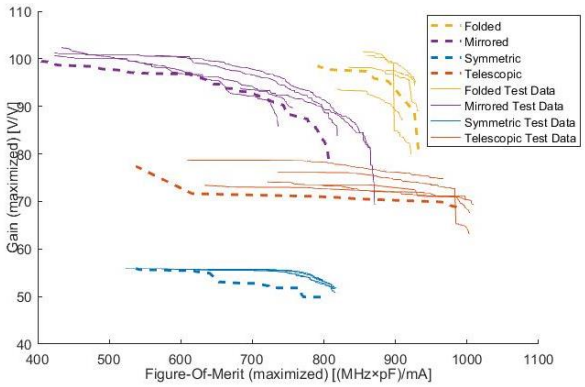

Fig. 8. Two objective optimization for Gain and FOM side-by-side with view of four objective pareto reformed into the same two objectives. The thicker dotted lines are plots of the four objective paretos, while the full lines are the two objective results.

A. Test Set

The test set had a similar setup as the library set. It was equal regarding circuits, metrics, netlists, input files, and GA parameters used. The optimizations were, performed for only two objectives at a time (1) Gain and FOM, (2) OS and VOS. These pairs were nominated due to the trade-off characteristic between them, competing with each other. They were arrested when there was convergence or when the population reached 6000 generations. These optimizations were performed 5 times for each objective pair, guaranteeing a greater level of statistical confidence.

In Fig. 8. and Fig. 9. the test set is represented side by side with their library equivalents. From these plots it is possible to see the curve similarities for each pair of objectives in the test set (the full lines), to the library optimizations, that were first subject to the Data Filtering and Pareto Reformation modules for the same two pairs (thicker, dotted lines). In Fig. 8. and Fig. 9. it is visible that the Gain and FOM, and OS and VOS results are as expected. This points towards the utility of the modules used.

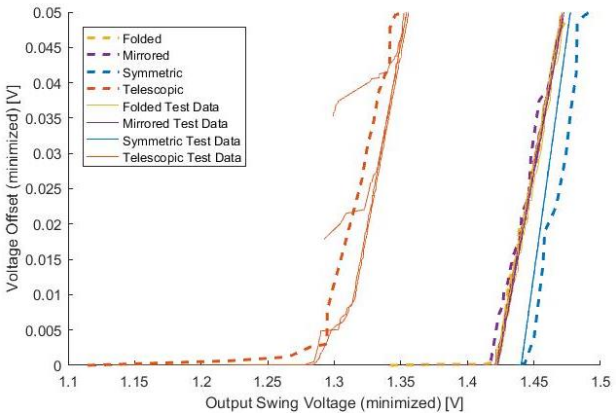

Fig. 9. Two objective optimization for OS and VOS side-by-side with view of four objective pareto reformed into the same two objectives. The thicker dotted lines are plots of the four objective paretos, while the full lines are the two objective results.

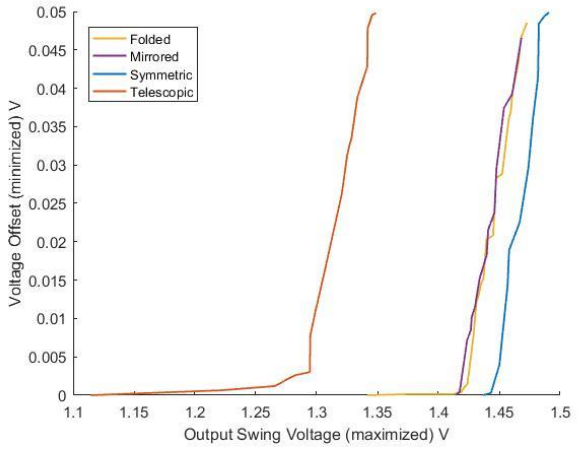

Fig. 10. 4 objective library reforned pareto for OS and VOS objectives.

Due to some data overlapping in Fig. 9. Two objective optimization for OS and VOS side-by-side with view of four objective pareto reformed into the same two objectives. The thicker dotted lines are plots of the four objective paretos, while the full lines are the two objective results. another plot is given with Fig. 10 that shows the library reformed pareto for the OS and VOS pair.

B. Model validation

To prove the idea of relationship between optimizations referred in the throughout this work, a quantitative analysis was performed. Essentially this program sums up the library to paretos, adds the line segments' closest points, and computes and sorts the distances.

A MATLAB® script was developed to iterate over all optimizations, and the simulations' output in them, entering them as input to the test program.

1) Closest topology

This approach relies exclusively on what was found to be the nearest library topology to the test individual by using as input (desired specifications), in the test program. If the selected topology matches that of the analyzed instance, then the similarity is verified. For each topology the success is measured through the rate of topology matching. Considering all the points the overall rate was:

$$total\ match\ rate = 76.46\ \%$$

The rate by objective pairs is displayed on Fig. 11.

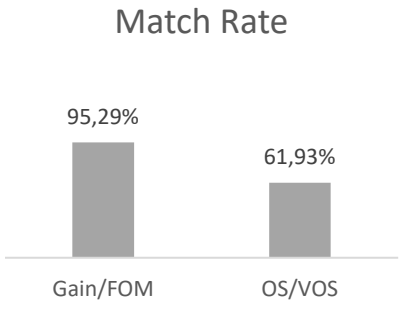

Fig. 11. Rate on the two objective pairs available (1) Gain/Fom and, (2) OS, VOS.

Portraying the significant difference in between the Gain/FOM and OS/VOS objective, with much higher proportion of test points that are closer to the same topology in the library. In Fig. 12. it is further detailed by topology per pair of objectives.

The low percentage values in the OS/VOS pair is a reflection of the similar performance of the Symmetric, Mirrored and Folded topologies, different from the Telescopic.

This was an indication that solely giving the closest topology was insufficient. Therefore, the average normalized distance to all library paretos was calculated.

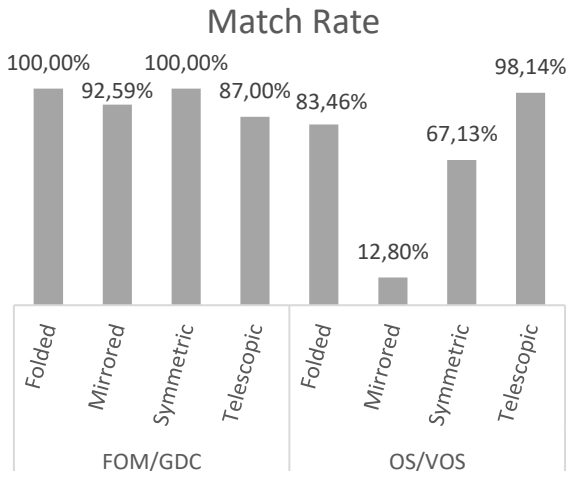

Fig. 12. Matching percentage in both objective pairs, per topology.

2) Average distance

The analysis of the average distance for VOS/OS provides valuable information, seen on Fig. 13., where it is made evident the similarities and differences just mentioned. This led to the inclusion of the normalized distances in the output of the program. In the case of the Folded, Mirrored and Symmetric topologies, if further optimizing the first ranked topology is not sufficient, attempting the other two is much more likely to result in a wanted outcome than trying the Telescopic-Cascode circuit.

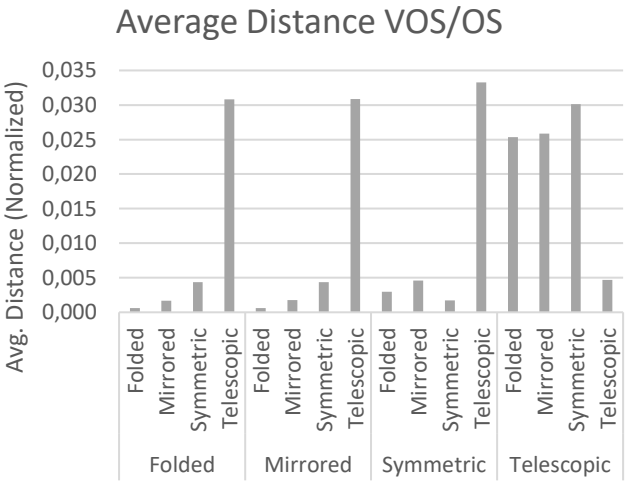

Fig. 13. Average distance grouped by topology of test points to library topologies for VOS/OS objective pair.

VI. Conclusions

A methodology for selection of best topologies for input goals, in both the cases of having and lacking feasible solutions was developed and implemented in a tool, as well as a library to integrate it. A test set to investigate the tool's efficacy was created and applied to it, analyzing the results.

It is possible to declare that the work was successful, delivering results that allow moderate optimism in considering the possibility of using the methods described as an electronic circuit assistant. This initial investigation into such tool requires further testing and adaptations to its in-market use.

A. Future work

This work introduces a method for topology selection that departs significantly from previous ones, and the tests that were done to it are also the initial trials into its accuracy. Thus, it would be beneficial to further introduce topologies and performance metrics that could further assess the capabilities and pitfalls of the tool's current form. When it comes to the measurements, commonly used and important ones like Area and Noise would be important to test for. As for the circuits, to widen the library to consider families of circuits such as, OpAmps, low-noise amplifiers and comparators could help confirm the tool's capacity to incorporate a wide variety of topologies.

This tool could also be added to the existing family of AIDA software, placing it as an optional feature before proceeding into sizing circuits. Furthermore, this tool could be integrated in a way that would take advantage of all the optimizations for which AIDA is used for, automatically storing everything required into the library. This could facilitate the acquisition of great amounts of information to not only create a complete library, but to also use this information to improve the selection tool. To prevent the tool from storing redundant circuits, an isomorphism algorithm could be added, such as the one seen in FEATS [22], which as the name indicates detects equal circuits or equivalent circuits.

Given that the pareto reformation module always stores the same information whenever a topology is deemed as non-feasible, which is represented by the index of the corresponding non-dominated solutions, then saving this information could save on the tool's computation time for a lot of cases, while occupying a small amount of memory.

References

[1] "WSTS," *World Semiconductor Trade Statistics (WTSS) Historical Billings Report*. [Online]. Available: https://www.wsts.org/esraCMS/extension/media/f/BBH/3959/bbhist-33.xls. [Accessed: 20-Apr-2019].

[2] "Gartner Market 2018," *Gartner Says Worldwide Semiconductor Revenue Grew 13.4 Percent in 2018; Increase Driven by Memory Market*. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-01-07-gartner-says-worldwide-semiconductor-revenue-grew-13-. [Accessed: 20-Apr-2019].

[3] L. Lavagno, I. L. Markov, G. Martin, and L. K. Scheffer, Eds., "Chapter 1: Design Flows," in *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology*, 2 edition., Boca Raton: CRC Press, 2016.

[4] "McClean Report Contents and Summaries 2019 | IC Insights." [Online]. Available: http://www.icinsights.com/services/mcclean-report/report-contents/. [Accessed: 27-Apr-2019].

[5] G. G. E. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proc. IEEE*, vol. 88, no. 12, pp. 1825–1854, Dec. 2000, doi: 10.1109/5.899053.

[6] M. G. R. Degrauwe *et al.*, "IDAC: an interactive design tool for analog CMOS circuits," *IEEE J. Solid-State Circuits*, vol. 22, no. 6, pp. 1106–1116, Dec. 1987, doi: 10.1109/JSSC.1987.1052861.

[7] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A Framework for Analog Circuit Synthesis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 8, no. 12, pp. 1247–1266, Jan. 1989, doi: 10.1109/43.44506.

[8] H. Y. Koh, C. H. Sequin, and P. R. Gray, "OPASYN: a compiler for CMOS operational amplifiers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 9, no. 2, pp. 113–125, Feb. 1990, doi: 10.1109/43.46777.

[9] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, "A mixed-integer nonlinear programming approach to analog circuit synthesis," in *[1992] Proceedings 29th ACM/IEEE Design Automation Conference*, 1992, pp. 698–703, doi: 10.1109/DAC.1992.227796.

[10] J. D. Lohn and S. P. Colombano, "A circuit representation technique for automated circuit design," *IEEE Trans. Evol. Comput.*, vol. 3, no. 3, pp. 205–219, Sep. 1999, doi: 10.1109/4235.788491.

[11] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 11, pp. 1237–1252, Nov. 2002, doi: 10.1109/TCAD.2002.804109.

[12] T. McConaghy, P. Palmers, M. Steyaert, and G. G. E. Gielen, "Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 557–570, Aug. 2011, doi: 10.1109/TEVC.2010.2093581.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[14] H. Li, X. Liu, F. Jiao, A. Doboli, and S. Doboli, "InnovA: A Cognitive Architecture for Computational Innovation Through Robust Divergence and Its Application for Analog Circuit Design," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 10, pp. 1943–1956, Oct. 2018, doi: 10.1109/TCAD.2017.2783344.

[15] K. Deb and D. Kalyanmoy, "Chapter 2: Multi-Objective Optimization," in *Multi-Objective Optimization Using Evolutionary Algorithms*, New York, NY, USA: John Wiley & Sons, Inc., 2001.

[16] C.-L. Hwang and K. Yoon, "Section III. 2.3.5. TOPSIS," in *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1981.

[17] R. F. S. Póvoa, J. C. da P. Goes, and N. C. G. Horta, *A New Family of CMOS Cascode-Free Amplifiers with High Energy-Efficiency and Improved Gain*. Springer International Publishing, 2019.

[18] K. Deb and D. Kalyanmoy, "Chapter 1: Prologue," in *Multi-Objective Optimization Using Evolutionary Algorithms*, New York, NY, USA: John Wiley & Sons, Inc., 2001.

[19] R. Povoa, N. Lourenco, R. Martins, A. Canelas, N. C. G. Horta, and J. Goes, "Single-Stage Amplifier Biased by Voltage Combiners With Gain and Energy-Efficiency Enhancement," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 65, no. 3, pp. 266–270, Mar. 2018, doi: 10.1109/TCSII.2017.2686586.

[20] R. Povoa, N. Lourenco, R. Martins, A. Canelas, N. Horta, and J. Goes, "Single-Stage OTA Biased by Voltage-Combiners With Enhanced Performance Using Current Starving," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 65, no. 11, pp. 1599–1603, Nov. 2018, doi: 10.1109/TCSII.2017.2777533.

[21] R. Povoa, N. Lourenco, R. Martins, A. Canelas, N. Horta, and J. Goes, "A Folded Voltage-Combiners Biased Amplifier for Low Voltage and High Energy-Efficiency Applications," *IEEE Trans. Circuits Syst. II Express Briefs*, pp. 1–1, 2019, doi: 10.1109/TCSII.2019.2913083.

[22] M. Meissner and L. Hedrich, "FEATS: Framework for Explorative Analog Topology Synthesis," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 2, pp. 213–226, Feb. 2015, doi: 10.1109/TCAD.2014.2376987.