http://dx.doi.org/10.5755/j01.eee.19.10.2464 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 10, 2013

# Sizing Analog Integrated Circuits by Current-Branches-Bias Assignments with Heuristics

I. Guerra-Gomez[1], E. Tlelo-Cuautle[2]
*[1]SEMTECH, Snowbush Mexico Design Center,*
*72000 Mexico*
*[2]INAOE,*
*Luis Enrique Erro No. 1, Tonantzintla, Puebla,*
*72840 Mexico*
*ivickguerra@gmail.com*

***Abstract*—This work shows the usefulness of assigning current-branches-bias levels, in order to improve and accelerate the sizing optimization of MOSFET-based analog integrated circuits (ICs). That way, the proposed procedure relies on the search of current branches from the associated incidence matrix by applying a recursive technique for exploring circuit graphs. The goal is focused on determining the bounds of the width/length (W/L) search space for each MOSFET before starting the sizing optimization process. As a case of study, the proposed current-branches-bias assignment (CBBA) approach is applied in the sizing optimization of the recycled folded cascode operational transconductance amplifier by applying evolutionary algorithms (EAs). From the feasible optimization results, we conclude that our proposed CBBA approach enhances and accelerates the biasing and sizing of analog ICs by EAs.**

***Index Terms*—MOSFET, operational transconductance amplifier, incidence matrix, topological circuit analysis, biasing.**

## I. INTRODUCTION

The sizing optimization of analog integrated circuits (ICs) by applying heuristic approaches has grown in the last decade [1]–[8]. For instance, evolutionary algorithms (EAs) are preferred because they provide a set of feasible solutions [4]. For EAs to work, one needs to specify: design variables, objectives, constraints, and search spaces. Besides, in IC design the search spaces in EAs can be reduced when paying attention on the specifications-design variables, e.g. the width (W) and/or length (L) of some MOSFETs can be the same or multiplier values of other MOSFETs [9], this usually happens in amplifiers composed of differential pairs and current mirrors [10].

Before sizing, biasing is performed to establish DC voltages/currents levels [11], [12], according to the available supply voltage(s) and current(s). Henceforth, this investigation introduces a DC current-branches-bias assignment (CBBA) approach to reduce the search spaces for Ws/Ls of MOSFET-based amplifiers to improve the sizing optimization of analog ICs with heuristics. In the next sections, we show the application of topological circuit analysis to search for current-loops. Afterwards, our proposed CBBA approach mirrors/distributes the DC current bias reference among all MOSFETs to determine the limits on the W/L search spaces.

The usefulness of our proposed CBBA approach is demonstrated by sizing the recycled folded cascode (RFC) operational transconductance amplifier (OTA) given in [9], by applying three EAs, namely: non-dominated sorting genetic algorithm (NSGA-II) [13], multi-objective EA with decomposition (MOEA/D) [14], and multi-objective particle swarm optimization (MOPSO) [15]. These EAs have been applied to sizing analog ICs in [1], [2], [4], [6]. In those references one can find their implementation details, and one can infer that at this moment any EA considers CBBA before starting the sizing optimization process.

For executing NSGA-II and MOEA/D, we apply two genetic operators: simulated binary crossover (SBX [16]) and differential evolution (DE [17]). The last section discusses the results provided by the three EAs with both genetic operators and with and without applying our proposed CBBA approach. Those results demonstrate that CBBA improves and accelerates the sizing optimization process of analog integrated circuits.

## II. TOPOLOGICAL CIRCUIT ANALYSIS

To search for current-loops, we appeal to formulate the incidence matrix **A** [11], [18], which is based on applying Kirchhoff's current law (KCL). That way, from a given circuit, a directed graph $\mathbf{G} = \langle \mathbf{N},\mathbf{B} \rangle$, is generated, where **N** is the set of nodes, and **B** the set of current branches. That way, matrix $\mathbf{A}_{kxl}$ has rows representing the nodes $\mathbf{N} = \{n_1,n_2,...,n_k\}$, and columns representing the branches (circuit elements) $\mathbf{B} = \{b_1,b_2,...,b_l\}$. Each element $a_{kl}$ in **A** can be 0, 1 or -1, according to topological rules [11], [18]:

- $a_{kl} = 1$ means that branch *l* leaves from node *k*,
- $a_{kl} = -1$ means that branch *l* enters to node *k*,
- $a_{kl} = 0$ means that branch *l* does not enter or leave node *k*.

In a circuit topology, some leaving branches can share just one entering current. For instance, by assuming that $i_a$ is distributed into $i_b$, $i_c$ *and* $i_d$, then $i_b=\alpha i_a$, $i_c=\beta i_a$ and $i_d=\gamma i_a$; where α, β and γ, are real positive numbers and their sum

Manuscript received November 14, 2012; accepted November 14, 2013.
This work is partially supported by CONACyT-Mexico under project 131839-Y.

equals one. By exploring matrix **A**, all current-loops can be found by executing the depth first search (**dfs**) algorithm shown in Algorithm 1 [19]. There is a vector named *Visited_flag* associating each branch to avoid visiting it twice.

```
Algorithm 1. Depth First Search (dfs) Algorithm
Require: branch b
1: Visited_flag[b] ← visited
2: for each branch $b_l$ adjacent to b do
3:     dfs($b_l$) if Visited_flag[$b_l$] ≠ visited
4: end for
```

The **dfs** algorithm explores all the adjacent branches for a given branch *b*. However, Algorithm 1 is modified to find the different current-distributions (Levels) in each branch. That way, we propose the top-down **dfs**TD algorithm shown in Algorithm 2, which requires: branch *b*, the upper node of *b*, namely *n*, and the bias level *CurLevel*. Vector *Visited_flag* associates a flag to each branch *b*, and vector *Bias_level* associates the bias level also to each *b*.

The **dfs**TD algorithm traverses the circuit in a top-down fashion, e.g. from the positive voltage bias ($\mathbf{V}_{dd}$) to the reference node or negative voltage bias ($\mathbf{V}_{ss}$). The first step is to mark branch *b* as visited one. The second line stores, in $\mathbf{B}_n$, the outgoing branches from *n* (different of *b*). Line 3 evaluates whether $\mathbf{B}_n$ is different from empty to subtract one to *CurLevel* for each $b_l$ in $\mathbf{B}_n$. If $\mathbf{B}_n$ is an empty set, there are not adjacent branches to *b*, and therefore it has the same level than its upper branch. In line 8, the level (*CurLevel*) is assigned to *b*. Line 9 sets the lower node of *b* to *n*. Line 10 finds the branches entering *n* and adds one to *CurLevel*. Line 11 repeats the procedure of line 2 (with the new node *n*). Finally, in line 12 there is a recurrence of the **dfs**TD to itself, if $b_l$ has not been visited. Processes in lines 2, 6, 9, 10 and 11, are performed by using matrix **A**, because it contains all the information about the circuit nodes, circuit branches, their connections and directions.

```
Algorithm 2. Top-Down dfs Algorithm (dfsTD)
Require: b, n, CurLevel
1: Visited_flag[b] ← visited
2: $\mathbf{B}_n$ ← set of outgoing branches from n (different of b)
3: if $\mathbf{B}_n \neq \emptyset$ then
4:     for each branch $b_l \in \mathbf{B}_n$ do CurLevel− = 1
5: else
6:     CurLevel ← level of the upper branch of b
7: end if
8: Bias_Level[b] ← CurLevel
9: n ← lower node of b
10: for each entering branch to n do CurLevel+ = 1
11:   $\mathbf{B}_n$ ← set of outgoing branches from n
12: for each branch $b_l \in \mathbf{B}_n$ do
13:     dfsTD($b_l$, n, CurLevel) if Visited_flag[$b_l$] ≠ visited
14: end for
```

## III. PROPOSED CURRENT-BRANCHES-BIAS ASSIGNMENT (CBBA) APPROACH

The goal of our proposed CBBA approach is focused on the distribution of the current bias reference(s) over all the leaving trajectories from the node assigned to the more positive supply voltage, e.g. $\mathbf{V}_{dd}$, to the node assigned to the reference or more negative supply voltage, e.g. $\mathbf{V}_{ss}$.

Algorithm 3 describes the distribution of currents: from a SPICE netlist, in line 1 matrix **A** is generated. From lines 2 to 5, the vector *Visited_flag* is initialized to control the recursive calls and vector *Bias_level*. The distributed or partitioned level is stored as a result of the auto-biasing process. In line 6 the outgoing branches from $\mathbf{V}_{dd}$, are stored in $\mathbf{B}V_{dd}$. Next, for each branch $b_l$ in $\mathbf{B}V_{dd}$ the *CurLevel* is initialized with zero, the method sets the zero level to $b_l$ and labels it as a visited branch. In line 11, the lower node of $b_l$ is stored in *n*, with the aim to build vector $\boldsymbol{B_n}$ formed by all outgoing branches from node *n* in line 12. In lines 13 to 15 there is a recursive call to **dfs**TD for each branch $b_n$, if it has not been already visited. At the end, each branch has an assigned level.

```
Algorithm 3. Distribution of Currents
Require: circuit netlist, specifying $V_{dd}$ and $V_{ss}$ nodes
1: Build matrix A and graph $\mathbf{G} = \langle \mathbf{N,B} \rangle$
2: for each branch $b_l \in \mathbf{B}$ do
3:     Visited_flag[$b_l$] ← not visited
4:     Bias_level[$b_l$] ← 0
5: end for
6: $\mathbf{B}V_{dd}$ ← set of branches outgoing from node $V_{dd}$
7: for each branch $b_l \in \mathbf{B}V_{dd}$ do
8: CurLevel ← 0
9: Bias_level[$b_l$] ← 0
10: Visited_flag[$b_l$] ← visited
11: n ← the lower node of $b_l$
12: $\mathbf{B}_n$ ← set of outgoing branches from n
13: for each branch $b_n \in \mathbf{B}_n$ do
14: dfsTD($b_n$, n, CurLevel) if Visited_flag[$b_n$] ≠ visited
15: CurLevel = 0
16: end for
17: end for
```

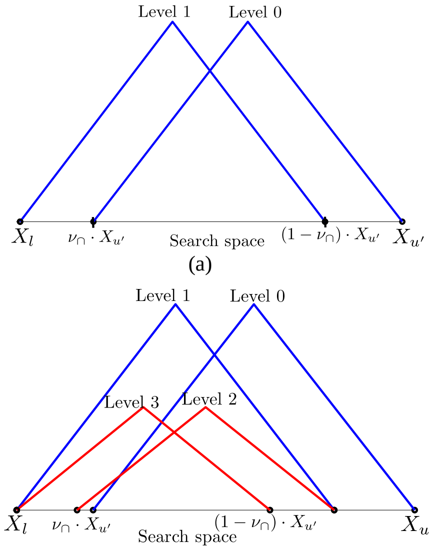
When each branch has a DC current bias level, another procedure sets the limits of the search spaces. It is done by a heuristic procedure. Let $\boldsymbol{X_l}$ be the lower and $\boldsymbol{X_u}$ the upper limits of the whole search space, $L_l^k$ the low limit and $L_u^k$ the upper limits in the search space for the *k*-th level. Algorithm 4 describes the assignment procedure. It consists of dividing the search space into sub-spaces according to the total number of levels (**TL**). The first step divides the entire search space into two parts corresponding to the two first levels (level 0 and level 1) that share an intersection region to relax the partitioning, and allowing exploring beyond the bound limits. The intersection between two levels (¬ą) depends on the total number of levels and is controlled by using an integer scaling factor (¡), as shown by (1). Since **TL** > 0 and ¡ > 0, then ¬ą < 0.5. In our experiments we used ¡ = 1

$$¬ą = (\mathrm{TL} + \chi)^{-1}. \tag{1}$$

The first loop in Algorithm 4 (lines 5 to 10) generates a result as the one shown Fig. 1(a). For the second loop, the process is repeated but this time the upper limit is bounded by $\mathbf{X}_{u'}$ as shown in Fig. 1(b). The algorithm continues until splitting the search space for all the levels. For an odd number of levels, Algorithm 4 assigns to the last level $\mathbf{X}_l$, its lower limit, and the last value of $\mathbf{X}_{u'}$ to its upper limit.

## IV. Application Example

Among the many available active devices [1], [5], [10], [20], [21], our CBBA approach is tested by optimizing the recycled folded cascode (RFC) operational transconductance amplifier (OTA) shown in Fig. 2. This RFC OTA has been already designed in [9], where key guidelines for manual design are provided there. It is worth mentioning that the authors clearly discuss on the difficulty of biasing and sizing that RFC OTA. Is for that reason that we apply heuristics herein for optimizing its performances, but the main goal is focused on showing that by applying our proposed CBBA approach, the execution of the three evolutionary algorithms (EAs): NSGA-II [13], MOEA/D [14] and MOPSO [15], is being improved and accelerated. The implementation details of those three EAs when applied to the optimal sizing of analog ICs can be found in [1], [2], [4], [6]. The inclusion of our proposed CBBA approach is straightforward but quite useful. In addition for the experiments, two genetic operators were used with NSGA-II and MOEA/D: simulated binary crossover (SBX [16]) and differential evolution (DE [17]).



(a)

(b)

Fig. 1. (a) First, and (b) second loops by executing Algorithm 4.

Table I shows the encoding of the RFC OTA that is biased with a DC current $I_{ref}$ = 400 µA and $\mathbf{V}_{DD}$ = 1.8 V. The target specifications, to be improved by performing optimal sizing using EAs and by applying our proposed CBBA approach, were taken from [9], which were obtained with a load capacitor of 5.6 pF. In our experiments, HSPICE simulations were performed in the optimization loops of the three EAs with a LEVEL 54 standard CMOS Technology of 90 nm.

*Algorithm 4.* Limit search space assignment procedure

```
Require: $X_l$, $X_u$, TL
k = 0
$X_{u'} = X_u$
¬ą = (TL + ¡ )$^{-1}$
while $k$ < TL do
    $X_l^k = X_{u'} * $ ¬ą
    $X_u^k = X_{u'}$
    $X_l^{k+1} = X_l$
    $X_{u'} = L_{aux} * (1 -$ ¬ą$)$
    $X_u^{k+1} = X_{u'}$
    k+ = 2
end while
if TL is odd then
    $X_l^{TL} = X_l$
    $X_u^{TL} = X_{u'}$
end if
```

The sizing of the RFC OTA is performed to optimize the eight target specifications, already provided in [9]: gain, gain-bandwidth product (GBW), phase margin (PM), input referred noise, input voltage offset, settling time (ST), slew rate (SR) and power consumption (PW). Equation (2) is used for this multi-objective sizing optimization problem as minimizing a vector function $\mathbf{f(x)}$. In this manner, $\mathbf{f(x)}$ is the vector formed by eight objectives: $f_1(x)$ = -1*Gain, $f_2(x)$ = -1*GBW, $f_3(x)$ = -1*PM, $f_4(x)$ = Input referred noise, $f_5(x)$ = Input voltage offset, $f_6(x)$ = ST, $f_7(x)$ = -1*SR, and $f_8(x)$ = PW.

$$\text{minimize } \mathbf{f(x)} = [f_1(x), f_2(x), \ldots, f_8(x)]^T$$
$$\text{subject to } h_l(x) \geq 0\ ,\ l = 1 \ldots p,$$

where

$$x \in X. \tag{2}$$

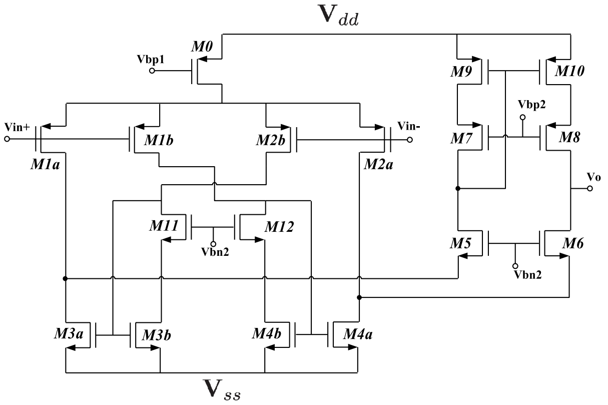


Fig. 2. RFC OTA.

In (2), $X : R^n$ | 0.18 µm ≤ $L_i$ ≤ 0.9 µm, 0.9 µm ≤ $W_j$ ≤ 130 µm, is the decision space for the ***n*** variables, and $h_l(x)$, with $l$ = 1...p, are the performance constraints. Additionally, we included the saturation condition in all transistors and the eight target specifications as constraints.

The three EAs: NSGA-II, MOEA/D and MOPSO, were executed with a population of 210 individuals along 250 generations. After performing our CBBA approach (Algorithm 4), the computed current-bias limits are shown in Fig. 3. Table II to Table IV show the sizing results provided by NSGA-II, MOEAD and MOPSO, by using the two genetic operators SBX and DE, and with and without applying our CBBA approach. In Fig. 4 to Fig. 6 are depicted the feasible solutions vs generations.

TABLE I. RFC OTA ENCODING.

| Gene | Variable | Transistors |
|---|---|---|
| $x_1$ | $L_1$ | M0,M3a,M3b,M4a,M4b,M9,M10 |
| $x_2$ | $L_2$<br>$2L_2$ | M5…M8<br>M1a,M1b,M2a,M2b |
| $x_3$ | $W_1$ | M0 |
| $x_4$ | $W_2$ | M1a,M1b,M2a,M2b |
| $x_5$ | $W_3$ | M3a,M4a |
| $x_6$ | $W_4$ | M3b,M4b |
| $x_7$ | $W_5$ | M5,M6 |
| $x_8$ | $W_6$ | M7,M8 |
| $x_9$ | $W_7$ | M9,M10 |
| $x_{10}$ | $W_8$ | M11,M12 |

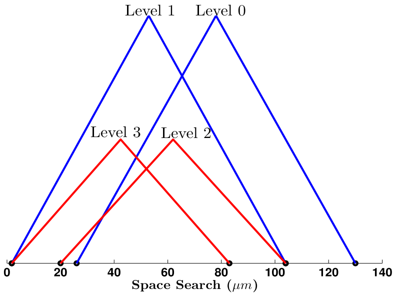


Fig. 3. Search space limits for the RFC OTA.

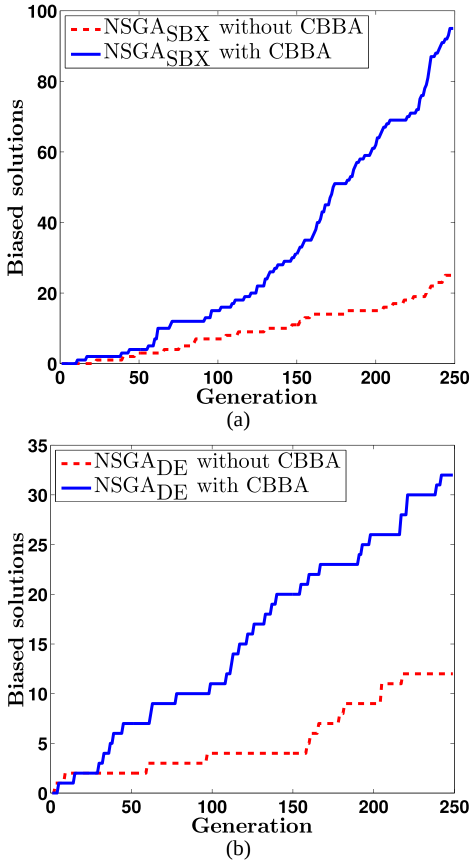
TABLE II. OPTIMAL FEASIBLE SOLUTIONS WITH NSGA-II.

| | Specs. | SBX | SBX | DE | DE |
|---|---|---|---|---|---|
| | | - | CBBA | - | CBBA |
| Gain dB | ≥55 | 65 | 61 | 61 | 64 |
| GBW MHz | ≥70 | 111 | 113 | 139 | 120 |
| PM deg | ≥65 | 62 | 68 | 65 | 75 |
| Offset mV | ≤11 | .15 | .30 | .51 | .28 |
| ST ns | ≤20 | 16.9 | 16.5 | 15.36 | 16 |
| SR V/μs | ≥48 | 80 | 120 | 97 | 87 |
| Noise $\mu V_{rms}$ | ≤69 | 70 | 69 | 68 | 68 |
| PW mW | ≤3.5 | 3.2 | 3.4 | 3.5 | 3.3 |
| **Variables** | | | | | |
| L1 μm | 0.5 | 0.23 | 0.18 | 0.18 | 0.21 |
| L2 μm | 0.18 | 0.24 | 0.18 | 0.19 | 0.18 |
| W1 μm | 64 | 129.98 | 129 | 87.19 | 75.66 |
| W2 μm | 32 | 126 | 82.15 | 127.12 | 78.39 |
| W3 μm | 12 | 98.66 | 100.46 | 53.02 | 20.8 |
| W4 μm | 4 | 41.49 | 47.77 | 17.45 | 72.42 |
| W5 μm | 8 | 14.05 | 13.64 | 10.17 | 9.34 |
| W6 μm | 32 | 65.3 | 82.06 | 128.53 | 66.84 |
| W7 μm | 32 | 6.42 | 6.04 | 65.56 | 15.6 |
| W8 μm | 4 | 7.03 | 2.96 | 29.58 | 46.91 |
| **Generation** | | 238 | 233 | 166 | 126 |

TABLE III. OPTIMAL FEASIBLE SOLUTIONS WITH MOEA/D.

| | Specs. | SBX | SBX | DE | DE |
|---|---|---|---|---|---|
| | | - | CBBA | - | CBBA |
| Gain dB | ≥55 | 62 | 63 | 63 | 66 |
| GBW MHz | ≥70 | 117 | 115 | 130 | 109 |
| PM deg | ≥65 | 50 | 69 | 58 | 69 |
| Offset mV | ≤11 | .12 | .19 | .19 | .2 |
| ST ns | ≤20 | 8.3 | 15.27 | 13.17 | 15.55 |
| SR V/μs | ≥48 | 81 | 92 | 80 | 88 |
| Noise $\mu V_{rms}$ | ≤69 | 84 | 72 | 73 | 70 |
| PW mW | ≤3.5 | 3.4 | 3.3 | 3.5 | 3.1 |
| **Variables** | | | | | |
| L1 μm | 0.5 | 0.58 | 0.21 | 0.18 | 0.21 |
| L2 μm | 0.18 | 0.24 | 0.18 | 0.26 | 0.18 |
| W1 μm | 64 | 111.6 | 120.95 | 80.51 | 75.66 |
| W2 μm | 32 | 65.77 | 83.12 | 128.22 | 78.39 |
| W3 μm | 12 | 77.47 | 88.59 | 111.68 | 20.8 |
| W4 μm | 4 | 16.21 | 35.14 | 33.37 | 72.42 |
| W5 μm | 8 | 13.36 | 13.65 | 9.41 | 9.34 |
| W6 μm | 32 | 78.99 | 75.58 | 76.47 | 66.84 |
| W7 μm | 32 | 28.42 | 6.65 | 8.52 | 15.6 |
| W8 μm | 4 | 40.25 | 6.73 | 19.31 | 46.91 |
| **Generation** | | 155 | 47 | 164 | 134 |

The sizing optimization process was performed along 5 runs for each EA with their corresponding genetic operators. As for any EA, all feasible solutions can be summarized in a Table showing statistics. That way, Table V shows the behavior of the three EAs with and without applying our proposed CBBA approach. It lists the average number of generations required to obtain biased feasible solutions, as well as the maximum and minimum number of generations for each EA with both genetic operators SBX and DE. Finally, Table VI shows the statistics for the non-dominated solutions in average for the 5 runs with and without CBBA after the sizing optimization process.



(a)

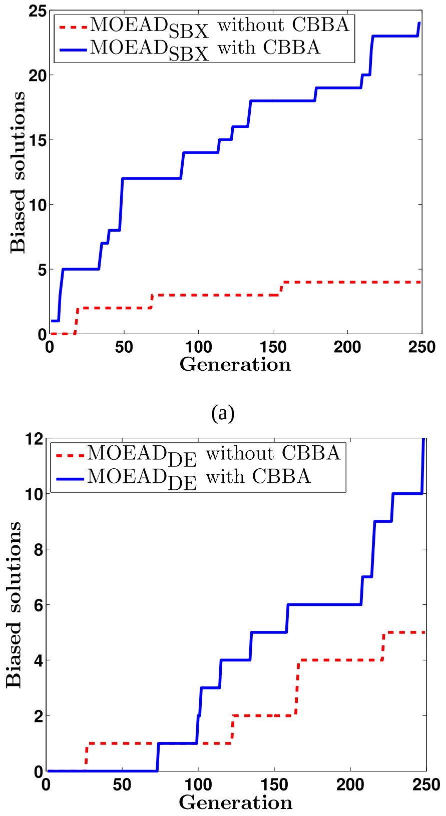
(b)

Fig. 4. Solutions with/without CBBA by NSGA-II.

TABLE IV. OPTIMAL FEASIBLE SOLUTIONS WITH MOPSO.

| | Specs. | Without CBBA | With CBBA |
|---|---|---|---|
| Gain dB | ≥55 | 62 | 67 |
| GBW MHz | ≥70 | 132 | 134 |
| PM deg | ≥65 | 52 | 68 |
| Offset mV | ≤11 | .36 | .25 |
| ST ns | ≤20 | 11.9 | 13.1 |
| SR V/μs | ≥48 | 87 | 90 |
| Noise $\mu V_{rms}$ | ≤69 | 70 | 70 |
| PW mW | ≤3.5 | 3.43 | 3.39 |
| **Variables** | | | |
| L1 μm | 0.5 | 0.18 | 0.27 |
| L2 μm | 0.18 | 0.23 | 0.2 |

| | Specs. | Without CBBA | With CBBA |
|---|---|---|---|
| W1 µm | 64 | 70.08 | 130 |
| W2 µm | 32 | 130 | 81.73 |
| W3 µm | 12 | 130 | 20.08 |
| W4 µm | 4 | 47.47 | 5.94 |
| W5 µm | 8 | 7.74 | 15.72 |
| W6 µm | 32 | 130 | 83.2 |
| W7 µm | 32 | 17.34 | 61.21 |
| W8 µm | 4 | 1.8 | 22.09 |
| **Generation** | | 122 | 39 |



(a)

(b)

Fig. 5. Solutions with/without CBBA by MOEA/D.

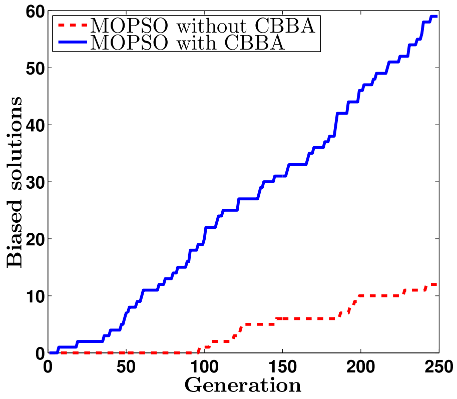


Fig. 6. Solutions with/without CBBA by MOPSO.

TABLE V. REQUIRED GENERATIONS TO BIAS ALL THE POPULATION.

| | Without CBBA | | | With CBBA | | |
|---|---|---|---|---|---|---|
| **Method** | **avg** | **max** | **min** | **avg** | **max** | **min** |
| **NSGA-II_SBX** | 107 | 150 | 40 | 100 | 150 | 24 |
| **NSGA-II_DE** | 130 | 150 | 69 | 109 | 150 | 52 |
| **MOEAD_SBX** | 147 | 150 | 143 | 110 | 120 | 103 |
| **MOEAD_DE** | 150 | 150 | 150 | 141 | 150 | 116 |
| **MOPSO** | 57 | 69 | 50 | 54 | 66 | 47 |

TABLE VI. NON-DOMINATED SOLUTIONS STATISTICS.

| Method | Without CBBA | With CBBA |
|---|---|---|
| **NSGA-II_SBX** | 92.6 % | 97.4 % |
| **NSGA-II_DE** | 89.6 % | 94.9 % |
| **MOEAD_SBX** | 85.5 % | 94.7 % |
| **MOEAD_DE** | 84.7 % | 95.9 % |
| **MOPSO** | 85.7 % | 98.2 % |

## V. DISCUSSION OF RESULTS

Applying heuristics like EAs in the sizing optimization of analog ICs, one cannot conclude on the superiority of one EA with respect to another one [4], [7], [22]. Besides, the genetic operators can improve their performance. But, one can improve and/or accelerate the optimization process by introducing better ways to set the search spaces, for instance. In this manner, we proposed a CBBA approach to reduce the search spaces in the optimal sizing of analog ICs, which provided good results. For instance, from the experimental results, NSGA-II with SBX (NSGA$_{SBX}$**)** improves all objectives with CBBA. However, the offset and PW exhibit slightly higher values, but all solutions accomplish the required target specifications from [9], as shown in the middle of Table II. For this case, an optimal solution without CBBA is found at generation 238, and with CBBA it is found five generations faster. However, an important thing is that by applying our CBBA approach the number of feasible solutions after 250 generations is higher (almost 4×), as shown in Fig. 4(a).

The right side of Table II shows the solutions provided by NSGA$_{DE}$. With CBBA the optimal solution exhibits improvement for gain, PM, offset and PW. GBW has a lower value due to the gain increase; the noise values and ST are similar, only there is a slight decreasing in SR. With CBBA, the optimal solution is found 40 generations faster and the solutions increase (almost 3×) as shown in Fig. 4(b). The middle of Table III shows the solutions provided by MOEA/D$_{SBX}$. With CBBA there is an improvement in gain, PM, SR, noise, PW, and the optimal solution is found more than 100 generations faster, and there are almost 5× solutions more than MOEA/D$_{SBX}$ without CBBA, as shown in Fig. 5(a). MOEA/D$_{DE}$ exhibits similar behavior, where the optimal solution is found 30 generations faster and the number of solutions is 2× when applying CBBA, as shown in Fig. 5(b).

Table IV lists the objective values for the feasible solutions provided by MOPSO. This time, applying CBBA achieves a general improvement for all the objectives, except for ST, that exhibits a slight increment. The optimal solution is found more than 80 generations faster. Figure VI shows an increase on the number of solutions after 250 generations (almost 5×).

In summary, the behavior of the three EAs by applying CBBA shows that they require in average less number of generations for biasing all the population, as shown by Table V. Notice that for NSGA-II_DE and MOEAD_DE the number of required generations in average is 15 % and 25 %, respectively, pretty less when CBBA is used. Regarding to the non-dominated feasible solutions, Table VI shows that the non-dominated solutions are greater when

CBBA is applied for all the EAs. On the other hand, the minimum improvement is around 5 % (for NSGA-II_SBX) and the best is more than 10 % (for MOEAD_DE and PSO).

## VI. CONCLUSIONS

A current-branches-bias assignment (CBBA) approach has been introduced in order to improve and accelerate the sizing optimization process of analog integrated circuits (ICs) composed of MOSFETs. The sizing was performed using three multi-objective evolutionary algorithms (EAs): NSGA-II, MOEAD and MOPSO, and they were executed by using SBX and DE as genetic operators.

The proposed CBBA approach executes a recursive depth first search in the associated incidence graph of the analog IC in order to find current-loops. Afterwards, the CBBA approach determines DC current bias levels with the aim to establish the bounds/limits of the W/L search spaces for each encoded design variable.

The proposed CBBA approach was tested by sizing an already designed RFC OTA. The results demonstrated the usefulness of the CBBA to accelerate the sizing optimization process through a reduction in the number of generations needed to guarantee convergence and to generate feasible solutions while improving/preserving the performances.

Quantitatively, the sizing optimization experiments by applying EAs showed a reduction up to 100 generations to find an optimal solution and an increase up to 5× in the number of generated feasible solutions when our proposed CBBA is executed. From different runs of the sizing optimization example, we found that when CBBA is used, it is expected to get all the population biased in fewer generations and also we showed that CBAA improves the number of non-dominated solutions. As a conclusion, our proposed CBBA approach is suitable to limit the search spaces for the design variables W/L, in order to enhance the sizing optimization of analog ICs. The gains are reflected in a reduction on the number of generations required to find an optimal solution and guaranteeing an increase in the number of non-dominated solutions, as shown by Tables V and VI.

## REFERENCES

[1] I. Guerra-Gomez, E. Tlelo-Cuautle, M. A. Duarte-Villasenor, C. Sanchez-Lopez “Analysis, Design and Optimization of Active Devices”, in *Integrated Circuits for Analog Signal Processing, E. Tlelo-Cuautle* (Ed.), 2012, Springer NY, pp. 1–30.

[2] S. Polanco-Martagon, G. Reyes-Salgado, G. Flores-Becerra, et al., “Selection of MOSFET Sizes by Fuzzy Sets Intersection in the Feasible Solutions Space”, *Journal of Applied Research and Technology,* vol. 10, no. 3, pp. 472–483, 2012.

[3] B. Liu, F. V. Fernandez, G. E. Gielen, “Efficient and Accurate Statistical Analog Yield Optimization and Variation-Aware Circuit Sizing Based on Computational Intelligence Techniques”, *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems,* vol. 30, no. 6, pp. 793–805, 2011.

[4] E. Tlelo-Cuautle, I. Guerra-Gomez, L. G. de la Fraga, G. Flores-Becerra, S. Polanco-Martagon, M. Fakhfakh, C. A. Reyes-Garcia, G. Rodriguez-Gomez, G. Reyes-Salgado “Evolutionary Algorithms in the Optimal Sizing of Analog Circuits”, in *Intelligent Computational Optimization in Engineering: Techniques & Applications*. M. Koeppen, G. Schaefer, A. Abraham (Eds.), 2011. Springer NY, vol. 366, pp. 109–138. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21705-0_5

[5] A. Chatterjee, M. Fakhfakh, P. Siarry, “Design of second-generation current conveyors employing bacterial foraging optimization”, *Microelectronics Journal,* vol. 41, no. 10, pp. 616–626, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.mejo.2010.06.013

[6] E. Tlelo-Cuautle, I. Guerra-Gomez, C. A. Reyes-Garcia, M. A. Duarte-Villasenor, “Synthesis of analog circuits by genetic algorithms and their optimization by particle swarm optimization”, in *Intelligent Systems for Automated Learning and Adaptation: Emerging Trends and Applications*, Raymond Chiong (Ed.). Information Science Reference: IGI Global. pp. 173–192, 2010.

[7] M. Barros, J. Guilherme, N. Horta, “Analog circuits optimization based on evolutionary computation techniques”, *Integration – The VLSI Journal,* vol. 43, no. 1, pp. 136–155, 2010.

[8] B. Smedt, G. Gielen, “WATSON: design space boundary exploration and model generation for analog and RF IC design”, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,* vol. 22, pp. 213–224, 2003. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2002.806598

[9] R. S. Assaad, J. Silva-Martinez, “The recycling folded cascode: a general enhancement of the folded cascode amplifier”, in *IEEE Journal of Solid-State Circuits,* vol. 44, no. 9, pp. 2535–2542, 2009. [Online]. Available: http://dx.doi.org/10.1109/JSSC.2009.2024819

[10] M. A. Duarte-Villasenor, E. Tlelo-Cuautle, L. G. de la Fraga, “Binary genetic encoding for the synthesis of mixed-mode circuit topologies”, *Circuits, Systems and Signal Processing,* vol. 31, no. 3, pp. 849–863, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00034-011-9353-2

[11] L. O. Chua, P. M. Lin, *Computer-aided analysis of electronic circuits*. Prentice-Hall NY, 1975.

[12] M. Tadeusiewicz, S. Halgas, “A Contraction Method for Locating All the DC Solutions of Circuits Containing Bipolar Transistors”, *Circuits, Systems and Signal Processing,* vol. 31, no. 3, pp. 1159–1166, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00034-011-9362-1

[13] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *Evolutionary Computation,* vol. 6, no. 2, pp. 182–197, 2002.

[14] Q. Zhang, H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Trans. on Evolutionary Computation,* vol. 11, no. 6, pp. 712–731, 2007. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2007.892759

[15] C. A. Coello-Coello, M. S. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization”, *IEEE Congress on Evolutionary Computation,* vol. 2, pp. 1051–1056, 2002.

[16] K. Deb, R. Agrawal, “Simulated binary crossover for continuous search space”, *Complex systems,* vol. 9, no. 2, pp. 115–148, 2005.

[17] R. Storn, K. Price, “Minimizing the real functions of the ICEC’96 contest by Differential Evolution”, in *IEEE Int. Conf. on Evolutionary Computation*, 1996, pp. 842–844. [Online]. Available: http://dx.doi.org/10.1109/ICEC.1996.542711

[18] Z. Qin, S. X.-D. Tan, C. K. Cheng, *Symbolic analysis and reduction of VLSl circuits,* Springer NY, 2005.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, Mc. Graw Hill, 2000, pp. 469–472.

[20] C. Sanchez-Lopez, F. V. Fernandez, E. Tlelo-Cuautle, S. X.-D. Tan, “Pathological Element-Based Active Device Models and Their Application to Symbolic Analysis”, *IEEE Trans. on Circuits and Systems I: Regular papers*, vol. 58, no. 6, pp. 1382–1395, 2011.

[21] L. F. Rahman, M. B. I. Reaz, M. A. M. Ali, et al., “Implementation of sense amplifier in 0.18-mu m CMOS process”, *Elektronika ir elektrotechnika (Electronics and Electrical Engineering),* no. 4, pp. 113–116, 2012.

[22] E. Tlelo-Cuautle, I. Guerra-Gomez, M. A. Duarte-Villasenor, L. G. de la Fraga, G. Flores-Becerra, G. Reyes-Salgado, C. A. Reyes-Garcia, G. Rodriguez-Gomez, “Applications of evolutionary algorithms in the design automation of analog integrated circuits”, *Journal of Applied Sciences,* vol.10, 2010.