Fayoum University Faculty of Engineering, 2025, Vol: 8(2), 55-63
https://fuje.journals.ekb.eg/
ISSN Online: 2537-0634
ISSN Print: 2537-0626

65-nm OTA Design Optimization Using Multi-Objective Genetic Algorithm

Mohamed M. Dakheel[1,*], Rania F. Ahmed[1], Ahmed H. Madian[2], Amr M. Gody[1]

[1]Electrical Engineering Department, Faculty of Engineering, Fayoum University, Fayoum 63514, Egypt
[2] NISC Research Center, Nile University, Cairo, Egypt

*Corresponding author: Mohamed M. Dakheel (mmm28@fayoum.edu.eg).

How to cite this paper: Dakheel, M.M., Ahmed, R.F., Madian, A H. & Gody A.M. (2025). 65-nm OTA Design Optimization Using Multi-Objective Genetic Algorithm, *Fayoum University Journal of Engineering, Vol: 8(2), 55-63*.
https://dx.doi.org/10.21608/fuje.2024.334034.1096

Abstract

This research paper examines the possible design optimization of analog circuits especially operational transconductance amplifier (OTA) using multi-objective genetic algorithm. All design criteria are developed and simplified and then used to run genetic algorithm on MATLAB. Final solutions located on the pareto front are examined to show various trad-offs between design parameters. A better more-advanced solution is also examined, where simulation software (CADENCE) is used to evaluate the individuals from each generation, and then results are fed to MATLAB to complete the algorithm. OCEAN script is used to perform simulations after netlist file is modified using results from MATLAB. Individuals are ranked based on their performance and high performing individuals are used in the next step. The best fitting solution is simulated using Cadence Virtuoso and all simulation figures are explained. The second approach is found to give better performance and meet the required objectives, this is due to the considering the device parameters' shift at different values for the circuit elements.

Keywords

OTA; Genetic Algorithm; Multi-objective Optimization; OCEAN script; Analog.

1. Introduction

Analog integrated circuits design is not an easy task to do, especially in circuits that have many requirements to be satisfied and even more parameters to determine. This means that designers are forced to abide by the trade-offs which causes accomplishment of some requirements and neglecting others.

One of the circuits that are the most difficult to design manually is OTA, with many requirements to accomplish, and yet more parameters. Designer has to carry multiple simulations before he can reach the near-optimal design he needs.

Using multi-objective optimization allows the designer to have a set of possible solutions with different results that can be examined and filtered based on preferred requirements.

Multiple optimization techniques have been examined over the last two decades; NSGA strategy has proved acceptable performance with much time needed to design a folded-cascode Op-AMP [1]. Particle Swarm Optimization (PSO) techniques also could be used to mimic the movement of bird groups, in which each parameter is considered as a particle, each particle has a history of positions, and needs to determine new position by adjusting its velocity, and by the knowledge of the food location [2]. More recent and widely used method is built upon Evolutionary Algorithm (EA). In EA, we mimic the evolution theory, where each step might be considered as a single reproduction of parameters that is tested using spice simulators and the result is fed back to the algorithm script (usually on MATLAB). These results are then used to evaluate the old population and only supreme population (Old and New) are used to reproduce new sets of parameters. After multiple rounds of applying the algorithm, we reach the optimal solution that can fit all our requirements [3].

There's also the Machine Learning (ML) based algorithms, which depends on training a network to predict the required design parameters. Numerous recent studies have adopted supervised learning models, primarily because of their dependence on labeled data. Regression and classification algorithms are prominently utilized, particularly tree-based models such as Random Forest and Decision Trees, in addition to Support Vector Regressor and gradient-based models. Nonetheless, there is a noticeable increase in the application of deep learning and Neural Networks [4]. However, these methods require some powerful machines with more experienced data sets to achieve a noticeable performance.

A few optimization techniques that worth mention is based on hybrid techniques developed recently like Artificial Bee Colony Optimization (ABCO), and Grey Wolf Optimization (GWO). Those algorithms have comparable performance to the Genetic Algorithm (GA), but with less median and standard deviation [5].

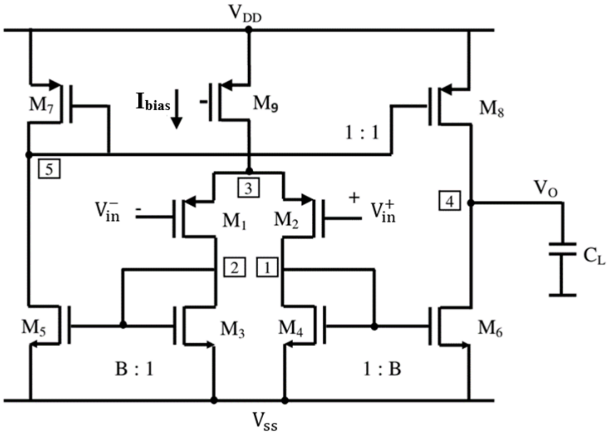

Figure 1 Symmetric OTA with Load Capacitance

Some literature introduces some surrogate-assisted optimization which speeds up the whole process of applying the Evolutionary Algorithm (EA) [6].

Among all optimization techniques, Genetic Algorithms (GA) remain the most reliable for applying without need to consume huge time for simulation on spice tools. It doesn't limit the script to specific topologies, the algorithm runs to accomplish required fitness functions supplied.

Multi-Objective Genetic Algorithm (MOGA) allows us to consider different objectives simultaneously, this in our case, allow us to minimize power consumption and area, at the same time, make sure gain and bandwidth are above certain limits.

Also, the time required for GA can be greatly reduced using the Rule-Guided Genetic Algorithm (RG-GA) which gives the algorithm specific set of rules to follow instead of blindly creating new population. For example, current value is related directly to the size of transistor.

Which means in case a current value is below the objective, we need to increase the width, and not consider thinner transistor widths. The RG-GA boosts the time performance by 1.5~3.3 times the GA [7].

This paper considers symmetric OTA to apply MOGA and calculate required dimensions of all transistors as well as biasing current. The symmetric OTA is widely used due to its better offset and CMRR performance [8].

2. Symmetric OTA

In this paper, we consider the design of 65-nm symmetric OTA with a capacitive load C_L (Shown in Figure 1). The main component of the circuit is differential pair (M1, M2), and three current mirrors (M3 – M5, M4 – M6, and M7 – M8), both input current mirrors have the same ratio B. We also have biasing current source transistor (M9).

The design parameters of our circuit are the dimensions (width and length) of all transistors, as well as the ratio of the current mirror (B).

For simplicity, we consider matching the input differential pair (M1 & M2) we also match (M3 & M4) and finally (M7 & M8). We also set length of transistors to a reasonable value according to 65-nm technology.

We then try to find the basic formulas to be optimized for gain and bandwidth.

The gain can be calculated simply,

$$A_{v_0} = \frac{v_o}{v_i} = \frac{i_6\, R_{N4}}{v_i} \tag{1}$$

Since matching,

$$i_1 = i_2 = g_{m1} v_i$$

And therefore,

$$A_{vo} = \frac{(B g_{m1} v_i R_{N4})}{v_i} = g_{m1} B R_{N4} \tag{2}$$

To calculate R_{N4} (Output resistance at node 4),

$$R_{N4} = r_{ds6} \parallel r_{ds8}$$

$$R_{N4} = \frac{1}{g_{ds6}} \parallel \frac{1}{g_{ds8}}$$

But,

$$g_{ds} = \lambda I_D$$

Where,

$$\lambda = \frac{\partial V_{DS}}{\partial L} = \frac{1}{V_E L}$$

Where V_E is a fitting parameter similar in concept to early voltage in BJT devices. Thus,

$$R_{N4} = \frac{1}{\frac{I_{D6}}{V_{E6}L_6}} \parallel \frac{1}{\frac{I_{D6}}{V_{E8}L_8}} = \frac{1}{I_{D6}} \cdot \frac{V_{E6}L_6 V_{E8}L_8}{V_{E6}L_6 + V_{E8}L_8} \tag{3}$$

Also,

$$g_m = \frac{\partial i_D}{\partial v_{DS}} = \sqrt{2\mu C_{ox}\left(\frac{W}{L}\right) I_D} = \sqrt{2k\left(\frac{W}{L}\right) I_D} \tag{4}$$

Substitute (3) and (4) into (2),

$$A_{vo} = \sqrt{2k_p}\, \frac{V_{E6}L_6 V_{E8}L_8}{V_{E6}L_6 + V_{E8}L_8} \sqrt{\left(\frac{W}{L}\right)_1} \frac{1}{\sqrt{I}} \tag{5}$$

We can also see that there's only one dominant pole in the circuit at node 4, resulting in bandwidth formula,

$$BW = \frac{1}{2\pi R_{N4} C_L}$$

And due to differential pair matching, we have $I_{D_6} = \frac{I}{2}$

$$BW = \frac{1}{4\pi C_L} \frac{V_{E6}L_6 + V_{E8}L_8}{V_{E6}L_6 V_{E8}L_8} IB \tag{6}$$

There are two more design requirements to consider, area and power dissipation.

For area we know that all routing and track widths are constant for the same technology, and therefore we can minimize are by controlling the dimensions of the transistors, so we can simplify the formula of the area by considering only the area of transistors.

$$Area = \sum_{i=1}^{n} W_i L_i \tag{7}$$

Where i is the order of the transistor and n is the number of transistors in the circuit (= 8) in our case, W and L are the width and length of i^{th} transistor.

The power dissipation can be calculated by multiplying the voltage across the circuit by the total current supplied to it.

$$P = (V_{DD} - V_{SS})(I + BI) = (V_{DD} - V_{SS})(1 + B)I \tag{8}$$

As we see in the equations above, we need to determine the value of current ratio to simplify the equations, this value could be limited if we considered the pole at node 5. As our OTA is single ended, so the capacitances on the other side than the output will be negligible for the Phase Margin [9].

Therefore, we have non-dominant frequency f_{nd},

$$f_{nd} \approx \frac{f_{T4}}{3+B} \tag{9}$$

According to [9], maximum B can be found by equating f_{nd} to $3 \times GBW$. Which results in $B = 5$. Many designers use typical value between 3 and 5.

3. Multi-Objective Genetic Algorithm Optimization

Any Multi-Objective Optimization Problem (MOP) can be described by the following set of equations,

Find a vector x that optimizes,

$$f(x) = \left(f_1(x), f_2(x), \dots, f_N(x)\right)^T \tag{10}$$

Subject to:

$$g_j(x) < 0, j = 1, \dots M;$$
$$h_h(x) = 0, k = 1, \dots K; \tag{11}$$
$$a(l) \leq x(l) \leq b(l), 1 \leq l \leq n$$

The first vector $f(x)$, represents a vector of objective functions that need to be maximized or minimized during the optimization problem.

Th second set of equation in (11) is called terms, first two $g_j(x)$ and $h_h(x)$ are the inequality and equality constraint functions, M and K are the number of constraints. Third equation represents the lower and upper boundaries for each gene $x(l)$.

As there's no unique solution for the optimization problem, Pareto optimality must be used to analyze the objectives [10].

A Pareto optimal solution is a solution in which an improvement in an objective requires the degradation of another. Optimal solution may be mapped to different objective vectors. Or to make it clearer, several optimal objective vectors representing different trade-offs between the objectives may exist [10].

We need a powerful global optimization technique to search the whole solution space. Genetic Algorithms are inherently robust and can efficiently search large solution spaces containing discrete or discontinuous parameters and non-linear constraints, without being trapped in local minima [10].

Figure 2 shows the flowchart for general GA optimization problem. The initial population consists of randomly generated individuals called "chromosomes". Next step is to evaluate each individual and quantify the result in a fitness value used to judge the quality of an individual. We then continue to create new generations by preserving supreme chromosomes and creating new ones through selection crossover and mutation. This means that only the best individuals reproduce while others are neglected, which improves the fitness of successive generations. This means that GA evolves until it finds a set of solutions where improvements tend to decay approaching no noticeable improvement yielding the Pareto front.

4. Analytical Expression Based Design

We consider four parameters to find, W_1,W_3,W_7, and I. We try to achieve the following specifications listed in Table 1.

Table 1 Design Requirements

No.	Specification	Desired Value
1	Gain	> 200
2	Bandwidth	$> 150\ kHz$
3	Layout Area	Minimal
4	Power Consumption	Minimal

Using the parameters from 65-nm process, as well as formulas derived earlier in section 2.

We write a MATLAB script considering $L_1 = L_2 = 0.36\mu m$, while for current mirrors and to minimize error in the mirrored current we choose higher channel length $L_3 = L_4 = L_5 = L_6 = 3.6\mu m$.We ran our script for a population size of 60 individuals, around 730 generations. We also set all other settings like lower and upper boundaries as well as crossover and selection sizes.

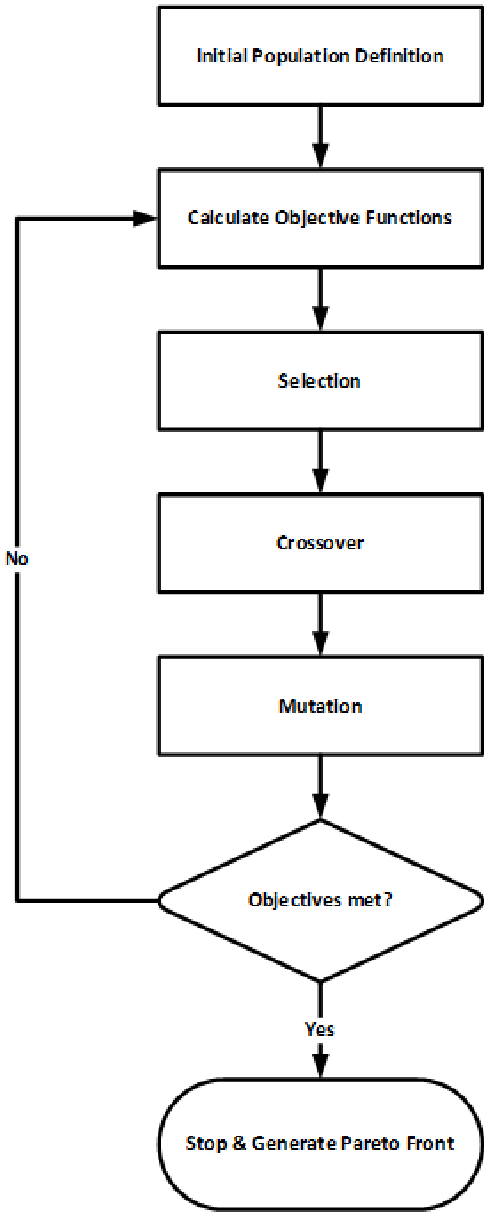
Initial Population Definition
Calculate Objective Functions
Selection
Crossover
Mutation
Objectives met?
No
Yes
Stop & Generate Pareto Front

Figure 2. Flowchart for General GA Optimization Problem.

The final Pareto front including 18 different individuals is shown in Table 2. We can see different objectives to consider trade-off between them.

As we notice from the Pareto front table, there are many individuals that meet our requirements, we selected an individual that not only meets both requirements but also have lower area and power consumption, individual No. 17 and carried the simulation using Virtuoso Cadence.

As seen in figure 3, the OTA with the proposed dimensions have gain of about 45 dB $\approx$180, and bandwidth of about 140 kHz.

Of course, this shift of results is expected as we're using same values to calculate the objective functions after each generation, not the simulated results from real model.

5. Using Circuit Simulation as Objective Criteria Evaluation

To have more accurate design, the new generation resulting from each mutation step is fed into simulation software to be evaluated. Based on this step a decision is reached to whether repeat the reproduction step, or to end the algorithm with this output.

We used a pre-installed Cadence version on a virtual machine, and MATLAB for Windows, so we setup a server (using ngrok) on the machine as well as on the Windows. Once MATLAB finishes the step it writes outputs to a file which is posted to the server.

On the virtual machine the server receives the file write component values to the OCEAN script, and then runs it. Once the simulation is completed, the results are sent to the server on Windows side where they are used for the evaluation step.

Figure 4 shows the basic steps for applying genetic algorithm with repetitive simulation using Cadence software.

We use the exact same procedure as used before, the only difference is that we don't evaluate the new generations using the objective functions, instead we use the results from the simulations to decide whether the output is acceptable or not.

Table 2 Pareto Front Individuals for first approach

Indiv.	W_1 (μm)	W_3 (μm)	W_7 (μm)	I_{bias} (μA)	**Gain**	**BW (KHz)**	**Area** (μm^2)	**Power** (μW)
1	1.24	1.99	0.95	0.744	181.6	210	22.17	8.93
2	1.22	2.09	0.96	0.66	191.3	180.23	22.25	7.92
3	1.106	3.03	3.38	0.547	200.1	151.4	46.98	6.57
4	1.107	2.617	1.47	0.549	199.8	151.7	30.288	6.58
5	1.107	2.634	1.64	0.548	199.9	151.4	31.56	6.57
6	1.106	3.04	3.14	0.547	200	151	45.33	6.57
7	1.106	3.15	3.79	0.547	200	151	50.76	6.57
8	1.12	2.256	0.98	0.563	198.4	155.5	24	6.76
9	1.106	2.994	2.68	0.547	200	151.01	41.66	6.57
10	1.106	2.94	2.53	0.547	200	151	40.19	6.57
11	1.24	1.99	0.956	0.75	180.9	207	22.168	8.93
12	1.236	2.0123	0.946	0.69	188.3	190.65	22.19	8.293
13	1.106	2.7268	1.79	0.547	200	151.01	33.367	6.574
14	1.108	2.459	1.13	0.549	199.9	151.69	26.68	6.59
15	1.1105	2.266	1.19	0.56	198	154.73	25.68	6.72
16	1.2339	1.99	0.96	0.749	180.59	206.96	22.168	8.94
17	1.106	2.983	1.98	0.547	200	151	36.53	6.57
18	1.1369	2.0368	0.97	0.58	196.99	160.26	22.45	6.96

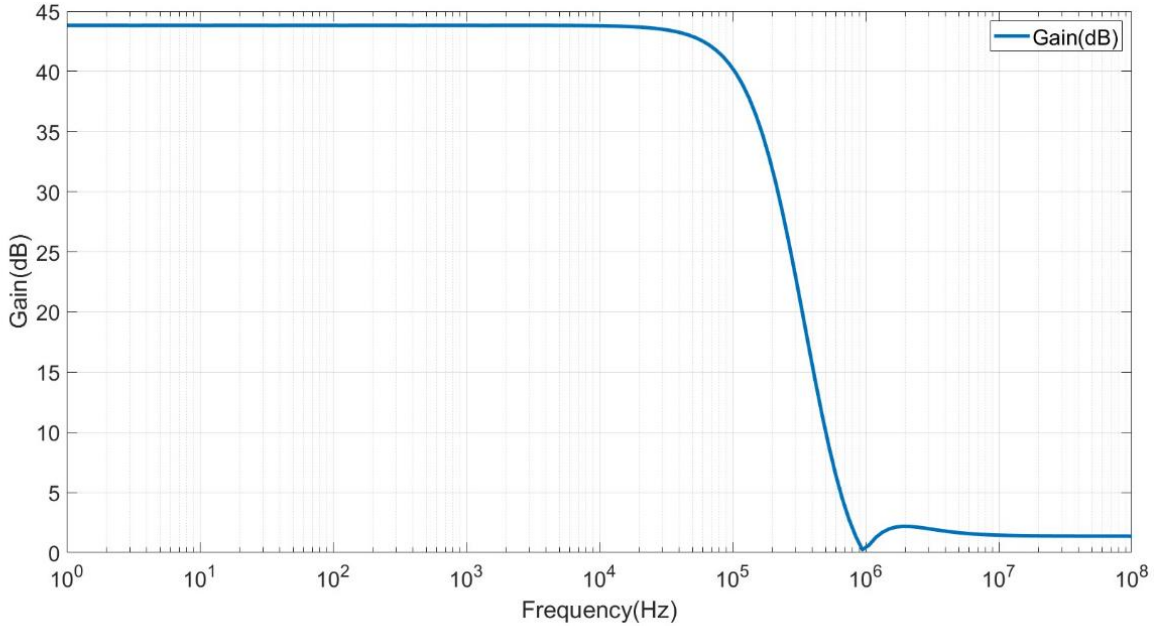

Figure 3 Frequency Response of The Designed Circuit

5.1. OCEAN Script

Open Command Environment for Analysis (OCEAN) is a programming language built upon SKILL language. It's used in Cadence Analog Design Environment (ADE) to perform circuit simulations from the command line [11].

Any circuit simulation performed using ADE can generate an OCEAN script which can be used to perform the simulation more than once.

We've written the OCEAN script to simulate the circuit once new values are written to the values file.

We modify the ocean script originally generated by ADE to follow our desired flow.

Here is the content of our OCEAN script which writes its output into a CSV file to be sent to MATLAB for individuals ranking.

```
simulator( 'spectre )
design("/home/cadence/simulation/Sim-
ple_OTA/spectre/schematic/netlist/netlist")
resultsDir(                    "/home/cadence/simulation/Sim-
ple_OTA/spectre/schematic" )
modelFile(
    '("/home/cadence/my_pro-
ject/tsmc65/tsmcN65/../models/spectre/toplevel.scs"
"tt_lib"))
analysis('ac ?start "1"    ?stop "10G"    ?dec "10"    )
envOption(
            'firstRun    t
        'analysisOrder    list("dc" "ac")
)
outPort = outfile( "/home/cadence/oceanresult.csv" "w");
fprintf(outPort "GainBW\tBW\n")
temp( 27 )
inPort = infile("matlabOut.txt")
simulator( 'spectre )
design("/home/cadence/simulation/Sim-
ple_OTA/spectre/schematic/netlist/netlist")
resultsDir(                    "/home/cadence/simulation/Sim-
ple_OTA/spectre/schematic" )
modelFile(
    '("/home/cadence/my_pro-
ject/tsmc65/tsmcN65/../models/spectre/toplevel.scs"
"tt_lib"))
analysis('ac ?start "1"    ?stop "10G"    ?dec "10"    )
envOption(
            'firstRun    t
```

```
envOption(
            'firstRun    t
        'analysisOrder    list("dc" "ac")
)
outPort = outfile( "/home/cadence/oceanresult.csv" "w");
fprintf(outPort "GainBW\tBW\n")
temp( 27 )
inPort = infile("matlabOut.txt")
```

5.2. Simulation and Results

To control the time required for our algorithm operation, we set the maximum allowable number of generations to 100, which will act as a termination criterion if no suitable solution is found during this time.

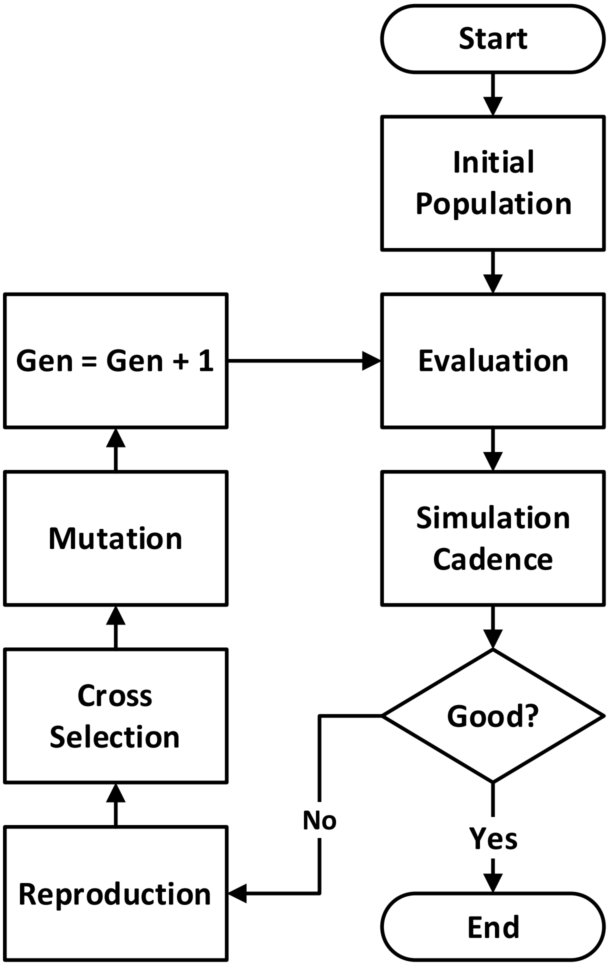

Figure 4 Simulation-Based Genetic-Algorithm Flowchart

The size of individuals per generation is set again to 60. So, we are required to run a total of (60×100=6000) simulations on Cadence. After completion, a pareto front of the best non-dominated individuals is fed to Cadence. We select the best 10 individuals as shown in table 3.

Figure 5 shows the pareto front for the two objectives against each other.

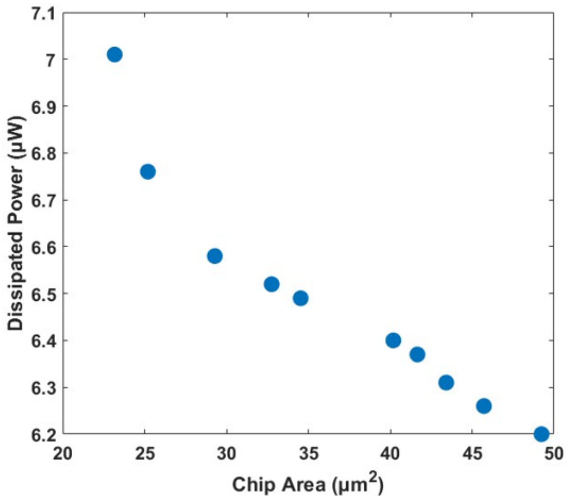

Figure 5 Pareto Front for The MOGA Problem

Again, we consider the best design which gives considerable low area and power and accomplishes both gain and bandwidth criteria. We choose individual No. 8.

The final gain and phase plots are shown in Figure 6.

As one might notice, this time algorithm's outputs match the simulation results, as we already base calculations upon results driven from Cadence.

To judge our circuit, we consider time required for the algorithm completion, the algorithm converge with saturated performance after 56 generations. This means a total of (56×60=3360) simulations. Total time required for the algorithm on a machine with an Intel Core-i7 processor with 4.7 GHz, is about 7.3 hours. Of course, this performance can be greatly boosted using a native software for the simulation (not through VMware).

6. Conclusion

This paper inspects the usage of evolutionary Mult-objective Optimization Genetic Algorithm (MOGA) to design an Operational Transconductance Amplifier (OTA) in the 65nm CMOS technology. A simple approach was used by concluding analytical expressions for the circuit's characteristics, and extracting the device parameters using CADENCE simulator, and then using MATLAB alone to work out a solution. Good but not optimal results were found. We then used more advanced technique, which included feeding the results from each generation from MATLAB to CADENCE and simulating those results to evaluate their performance. The second method gave optimal solutions meeting all required objectives.

Table 3 Pareto Front Individuals for Second Approach

Indiv.	W_1 (μm)	W_3 (μm)	W_7 (μm)	I_{bias} (μA)	**Gain**	**BW (KHz)**	**Area** (μm^2)	**Power** (μW)
1	1.106	2.994	2.68	0.531	200	150.2	41.66	6.37
2	1.13	2.245	0.92	0.563	199.4	156.4	25.2	6.76
3	1.106	3.13	3.82	0.522	200.2	148.4	45.72	6.26
4	1.107	2.717	1.57	0.549	199.8	151.7	29.288	6.58
5	1.106	2.94	2.53	0.534	200	151	40.19	6.4
6	1.106	3.09	3.04	0.526	200	150.2	43.42	6.31
7	1.106	3.24	3.68	0.518	199	151	49.24	6.21
8	1.104	2.974	1.88	0.541	200	150.8	34.53	6.49
9	1.34	1.95	0.98	0.585	188.6	205	23.17	7.01
10	1.102	2.465	1.66	0.544	199.9	150.6	32.76	6.52

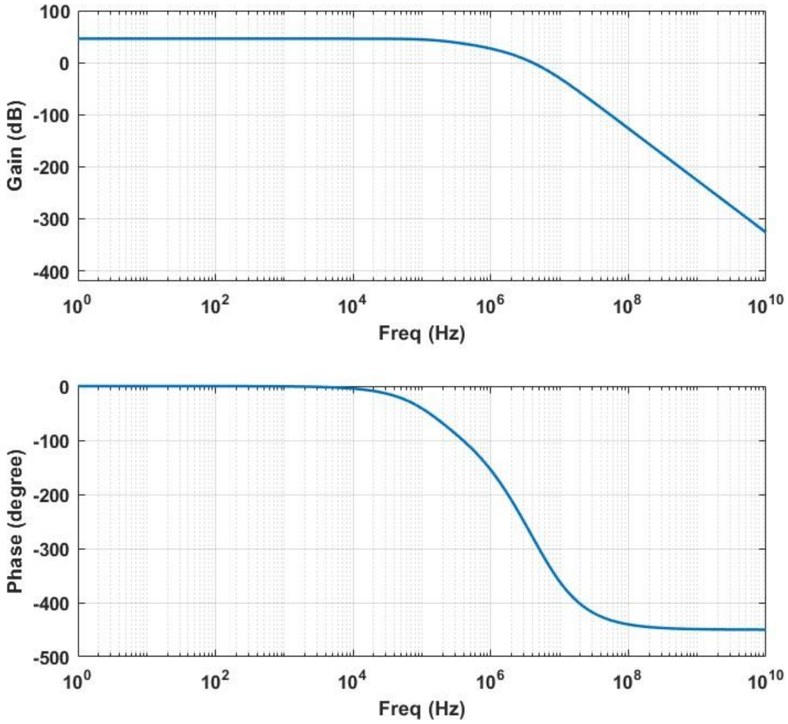

Figure 6 Gain and Phase of Simulated OTA

References

[1] Takhti, M., Beirami, A., Shamsi, H.: Multi-Objective Design Automation of The Folded Cascode OP-AMP Using NSGA-II Strategy. In: International Symposium of Signals, Circuits and Systems, Iasi, pp. 1–4 (2009).

[2] Mansour Barari, Hamid Reza Karimi, Farhad Razaghian, "Analog Circuit Design Optimization Based on Evolutionary Algorithms", Mathematical Problems in Engineering, vol. 2014, Article ID 593684, 12 pages, (2014).

[3] Qiang Long, Changzhi Wu, Xiangyu Wang, Lin Jiang, Jueyou Li, "A Multiobjective Genetic Algorithm Based on a Discrete Selection Procedure", Mathematical Problems in Engineering, vol. 2015, Article ID 349781, 17 pages, (2015).

[4] Attaoui, Y., Chentouf, M., Alaoui Ismaili, Z.E., & Elmourabit, A. “Machine Learning in VLSI Design: A Comprehensive Review”. Journal of Integrated Circuits and Systems (2024).

[5] Rashid, R., Raghunath, G., Badugu, V., & Nambath, N. “Performance evaluation of evolutionary algorithms for analog integrated circuit design optimization”. Microelectronics Journal (2023).

[6] Yin, S., Wang, R., Zhang, J., Liu, X., & Wang, Y. Fast surrogate-assisted constrained multiobjective optimization for analog circuit sizing via self-adaptive incremental learning. “IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems”, 42(7), 2080-2093 (2023).

[7] Zhou, R., Poechmueller, P., & Wang, Y. “An Analog Circuit Design and Optimization System with Rule-Guided Genetic Algorithm”. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41, 5182-5192. (2022).

[8] Sedra, A.S., Smith, K.C., Microelectronic Circuits, International Sixth edn. Oxford University Press, Oxford (2011).

[9] Sansen, W.M.C., Analog Design Esentials. Springer, Heidelberg (2006).

[10] Oltean, G., Hintea, S., Sipos, E.: Analog Circuit Design Based on Computational Intelligence Techniques. Journal of Automation, Mobile Robotics & Intelligent Systems 3, pp. 63–69 (2009).

[11] OCEAN Reference, Cadence. San Jose, California, USA, June (2004).