



# An Efficient Three-Stage Yield Optimization Technique for Analog Circuits Using Evolutionary Algorithms

Abbas Yaseri<sup>1</sup>, Mohammad Hossein Maghami<sup>2\*</sup>, Mehdi Radmehr<sup>1</sup>

1) Department of Electrical Engineering, Sari Branch, Islamic Azad University, Sari, Iran

2) Faculty of Electrical Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran

abbas.yaseri@gmail.com; mhmaghami@sru.ac.ir; maradmehr@gmail.com

Received: 2020/11/28; Accepted: 2021/01/07

## Abstract

In addition to the improved calculation of the parameter values, a high yield estimation is necessary for designing analog integrated circuits. Although Monte-Carlo (MC) simulation is popular and precise for yield estimation; however, its efficiency is not high enough and it requires too many costly transistor-level simulations. Therefore, some accelerated methods are needed for MC simulations. This paper presents a novel approach for improving automated analog yield optimization using a three-stage strategy. Firstly, critical solutions are recognized using Critical Analysis (CA) and Multi-objective Optimal Computing Budget Allocation (MOCBA). Then they are separated from non-critical answers. It's so helpful to avoid repeating the Monte Carlo (MC) simulations of non-critical solutions. Due to the existence of several objective functions (typically more than one) in the yield optimization problem, by using the Multi-Objective Optimization (MOO) in the second stage, more precise answers can be found. Finally, MC simulations are performed to explore the proposed algorithm performance. Simulation results show that our approach locates higher quality in terms of yield rate within less run time and without affecting the accuracy.

**Keywords:** Yield optimization, MOCBA, Monte-Carlo, Critical Analysis, Multi-Objective Optimization

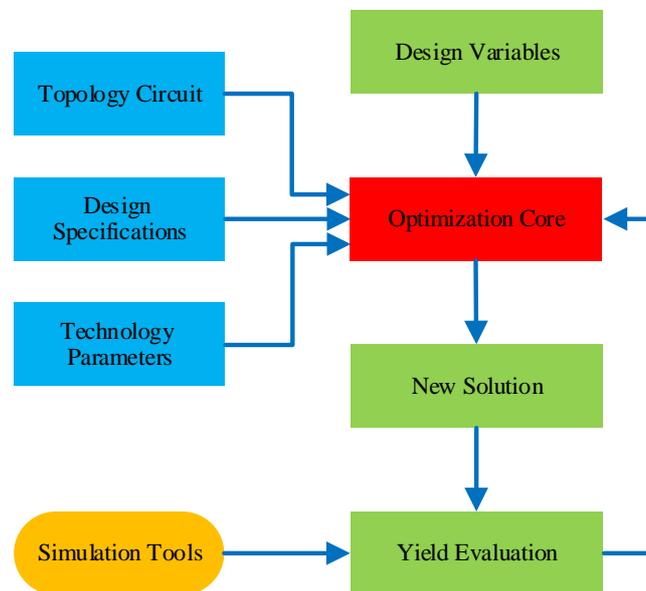
## 1. Introduction

Yield, in the integrated circuit manufacturing industry, is defined as the ratio of the number of items produced that are qualified for sale to the total number of items produced [1]. Design for Manufacturability and Yield is one of the most important concepts in analogue integrated circuits manufacturing [24]. Novel and robust techniques for automated analog yield optimization are urgently needed to avoid costly re-design iterations. However, identifying an optimized nominal design solution faces vast challenges.

The general flow chart for yield optimization is given in Figure 1[3]. The optimization engine generates a design solution, and the evaluator estimates its yield. Since an accurate yield estimation is much more computational expensive than a nominal performance evaluation, such a yield optimization often takes a large amount of computing resources and CPU time. In order to improve the computational efficiency of yield optimization, two approaches can be considered:

- 1- cut down the number of yield estimations.
- 2- reduce the computational costs of yield estimations.

Most of the existing yield optimization techniques put their effort into the latter approach. The yield analysis approaches can be classified into statistical methods and non-statistical methods. Non-statistical methods include corner-based methods [4] and performance-specific worst-case design (PSWCD) methods [5]. The main advantage of these methods is the limited number of their simulations. But their drawback is the difficulty to know in advance the worst-case corner points. Moreover, simple and fast sensitivity analysis in worst-case design methods may harm the accuracy in nanometer technologies. Statistical methods include Monte-Carlo (MC)-based methods and response-surface-based (RSB) methods [6]. The challenge of these methods is considering the balance between the accuracy and the complexity of the model, as well as the accuracy and the number of samples. Hence, these approaches can be used for yield optimization but have considerable accuracy problems.



**Figure 1. General flowchart for yield optimization [2]**

Monte Carlo-based (MC-based) yield optimization methods feature high generality and accuracy, so they are still the most reliable and commonly used technique for yield estimation till now. Nevertheless, a large number of simulations are needed for MC analysis, therefore limiting its use within an iterative yield optimization loop. Although some speed enhancement techniques for MC simulation have been proposed [7,8], the efficiency gain is not high enough to make MC-based yield optimization practical.

Therefore, in this paper, a three-stage approach is proposed for the yield optimization of analog integrated circuits. This method improves the efficiency of yield optimization by cutting down both the number of simulation and computational budget of yield estimations. Due to its high efficiency in terms of function evaluations, the proposed approach is quite suitable for yield optimization problems.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge for the proposed algorithm. The general framework of proposed approach is given in Section 3. Section 4 tests proposed algorithm by practical examples. The concluding remarks are presented in Section 5.

## 2. Background knowledge for the proposed method

Since yield optimization at this research is established based on MOCBA, MOO, and MC, they are briefly reviewed here.

### 2.1 Multi-objective computing budget allocation

Optimal computing budget allocation (OCBA) is a popular ranking and selection (R&S) procedure. The framework is proposed to enhance the simulation efficiency by intelligently allocating replications to each alternative solution based on mean and variance. OCBA shows that the computing budget allocated to any non-best design is proportional to its variance and inversely proportional to its difference from the best design [9]. OCBA problem is formulated as an optimization model with the objective of maximizing the probability of correct selection (PCS), which is the probability that the design we select is truly the best. The problem is subject to the constraint of limited computing budget, and the decision variables are the number of replications allocated to each design alternative. Therefore, there are two key issues in solving OCBA problems:

- 1- how to define PCS
- 2- how to solve the non-linear optimization model.

Usually, PCS is approximated by a proper lower bound instead of a closed-form analytical expression [10]. Even though the fundamental OCBA framework is proposed for selecting the best alternative with just one objective, OCBA procedures for solving multi-objective problems have been developed. For MOCBA problems, direct cardinal comparison may not be applicable as multiple objectives may compete with one another and we are not able to find a single best solution that simultaneously optimizes all the objectives. The transformation approach, which converts multi-objective problems into single-objective ones by aggregating the performance measures using a functional form is indifferent from the fundamental single-objective problems, and there is no consensus about the value of the weights [11].

The concept of Pareto optimality is employed instead, where the goodness of a design is measured in terms of domination [12-14]. Because of the underlying complexity in cross comparisons of designs, all designs can be grouped by the roles they are playing, either dominating or being dominated. For designs playing the role of dominating, each of them is dominating multiple designs, and thus multiple comparisons are incurred, the related allocation follows the sum of weighted variance rule, where the variance rules out the comparisons for significant differences; for designs playing the role of being dominated, each design is only dominated by one design.

### 2.2 Critical analysis

MC simulation is a common method for estimating of yield; however, practically it is not very effective because of its slowness. One way to alleviate this problem is the removal of non-critical solutions which prevents excessive allocation of simulations and computational efforts [15]. A large number of simulations repeats to considerably discern between competing designs as an important contribution preventing optimization of random simulation. The cost of computing, which itself involves repetition of simulation, is expensive and usually unacceptable. To ensure the correctness of the optimum selection of an optimal plan, a larger amount of computational budget should be disbursed for critical designs to determine the best plan.

Meanwhile, limited computational effort should be considered for non-critical projects that have little chance of representing the best of the project.

Based on this theory, by using the MOCBA technique, critical and non-critical solutions can be distinguished. Also, simulation repetitions could be allocated between them optimally. Therefore, it reduces computational effort by allocating optimal computing budget between solutions. Correct selection with a rational yield estimation while have the smallest computational effort is the main requirement. To reduce the estimator variance, a greater number of simulations should be allocated to the critical solutions which is the main aim to use MOCBA in this approach. This will cause less computational effort spent on non-critical solutions that have little impact recognition the suitable solutions, even with large variances. This is achieved by utilizing of Algorithm 1 [16].

**Algorithm 1. MOCBA for CA**

- 1- Run  $N_0$  replications for each design. Set the iteration index  $v=0$ , and individual scenario sample sizes  $N_1^v, \dots, N_n^v = N_0$ .
- 2- Construct the observed Pareto set  $S_p^{obs}$ .
- 3- While (termination condition is not satisfied) Increase the number of simulation replications by a certain amount  $D$ .
- 4- Calculate the new allocation  $N_1^{v+1}, \dots, N_n^{v+1}$  according to the allocation rules stated in Lemma 4, 5 that can be found in [16].
- 5- Perform an additional  $\min(t, \max(0, N_i^{v+1} - N_i^v))$  replications for design  $i = 1, 2, \dots, n$ . Set  $v = v + 1$ .
- 6- Construct a new observed Pareto set  $S_p^{obs}$ .
- 7- Output designs in the observed Pareto set  $S_p^{obs}$ .

$t$  is the maximum possible number of replications that can be allocated to a design at each iteration.

**2.3 Multi-objective optimization**

The accuracy and reliability of the results can be satisfied by implementing two or more objective functions that are called multi-objective optimization (MOO) [22-23]. The basis of multi-objective analog circuit optimization is multi-objective evolutionary algorithms (MOEAs) [19-22]. Here is a compromise between the incompatible objectives. In MOO, instead of the best solution, there are several solutions for all purposes. The multi-objective optimization equation is given by

$$\text{Min/max } f_1(x), f_2(x), \dots, f_n(x) \tag{1}$$

Subject to:  $x \in U$

Where  $n$  is the number of objective functions,  $U$  is feasible set,  $f_n(x)$  is  $n_{th}$  objective function,  $x$  is solution and min/max is combined object operations.

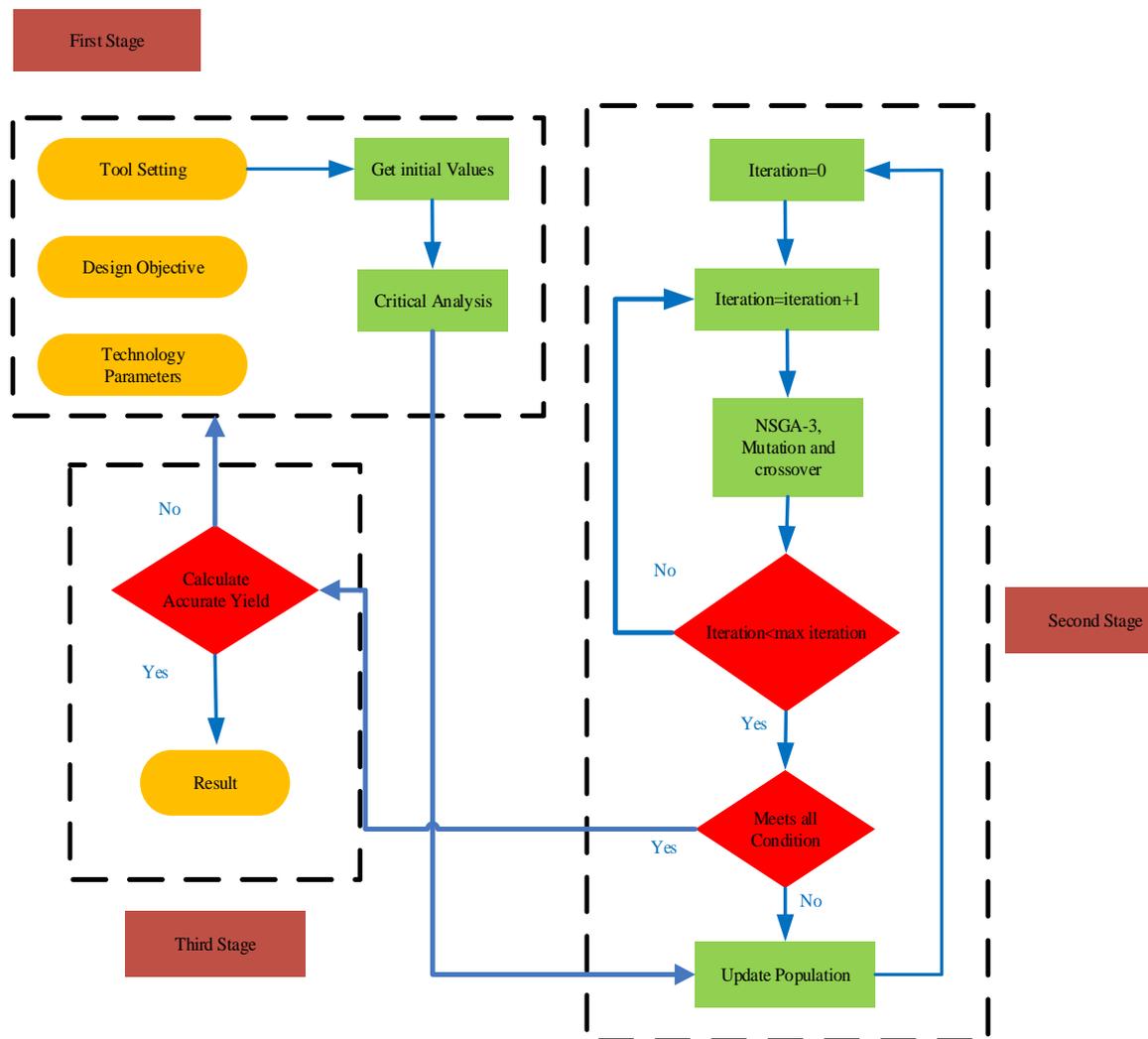
**3. Proposed approach**

Figure 2 shows the flowchart of the proposed approach that is composed of a three-stage algorithm. At the first stage, a set of designs under defined constraints is

generated. Then, the designs that met our desired characteristics, are separated from non-critical designs by the use of CA and MOCBA as previously described. Then the selected best designs are forwarded to the next stage. For optimizing the selected solutions, Non-dominated Sorting Genetic Algorithm-III (NSGA-III) is used at this stage [23].

**3.1 NSGA - III**

We expose the NSGA - III that is designed to face up with many objectives at the same time (more than two). The NSGA - III is based on the steps described in algorithm 2:



**Figure 2. The flowchart of the proposed approach**

**Algorithm 2. NSGA – III**

- 1: Calculate the number of reference points (H) to place on the hyper-plan
- 2: Generate the initial population at random taking into account the resources assignment constraints (POP chromosomes)

- 3: Realize the non-dominated population sorting
- 4: for  $i = 1$  Stopping criteria do
- 5: Select two parents P1 and P2 using the tournament method
- 6: Apply the crossover between P1 and P2 with a probability  $P_c$
- 7: Realize the non-dominated population sorting
- 8: Normalize the population members
- 9: Associate the population member with the reference points
- 10: Apply the niche preservation (counter)
- 11: Keep the niche obtained solutions for the next generation
- 12: end

Determination of reference points on a hyper-plane: We must define a set of reference points to ensure the diversity of the obtained solutions. Different points are placed on a normalized hyper-plan that have the same orientation in the all the axis. The number of reference points ( $H$ ) is defined by:

$$H = \frac{C + g - 1}{C} \quad (2)$$

Where  $C$  is the number of objective functions,  $g$  is the number of divisions to consider on every objective axis (for 3 objectives and 4 divisions, we will have 15 reference points). The reference points are place on the hyper-plan and the solutions will be described by a Pareto front, then the solutions will be associated with the created reference points.

Normalization of the population members: adaptive normalization of population members is applied and the arbitrary point is specified by recognizing the minimum value ( $Z_i^{\min}$ ).  $f_i'(x)$  translates each objective value of  $S_i$  in such a way that the arbitrary point of translated  $S_i$  becomes a zero vector.

$$f_i'(x) = f_i(x) - Z_i^{\min} \quad (3)$$

Association among reference points and solutions: After normalizing each objective function, it is necessary to associate each population member with a reference. We define a reference line for each point joining the reference point with the origin point. Then, we determine the perpendicular distance among each population member and each reference line. Finally, the reference point that has the closest reference line from a population individual is associated to this population member.

Niche preservation operation: A reference point can be associated to one or more solution members, but we must keep the solution that is closer of the point (perpendicular distance from the reference line).

Genetic operators: The children generation has been made applying the genetic operators used in the NSGAI algorithm. we have fixed a population size (POP) close to the number of reference points ( $H$ ) to give the same importance to all the population members.

Finally, by using the selected solutions from the previous step and the MC method, the accurate yield is calculated. If the desired value is not obtained, stages 1 and 2 of the algorithm are repeated. The steps of the three-stage presented approach are below:

### Algorithm 3. Proposed Approach

Step 1: A set of designs under defined constraints is generated.

Step 2: Critical designs that meet our desired specifications are separated from non-critical designs by the use of CA.

Step 3: For more precise solutions, NSGA-III is performed on selected solutions to optimize designs.

Step 4: MC method is used to calculate the exact yield for the selected solution.

Step 5: If the desired value is not obtained, go to step 1.

Step 6: End.

#### 4. Simulation Results

The proposed algorithm is tested on a two-stage fully differential op-amp shown in Figure 3. All simulations are performed on a workstation with an Intel(R), Core (M) i7-4790K CPU@4GHz, 16GB RAM, and 64-bit operating system with the x64-based processor. MATLAB R2020 and Synopsys HSPICE are used as the MC simulator and evaluator of circuit performance parameters.

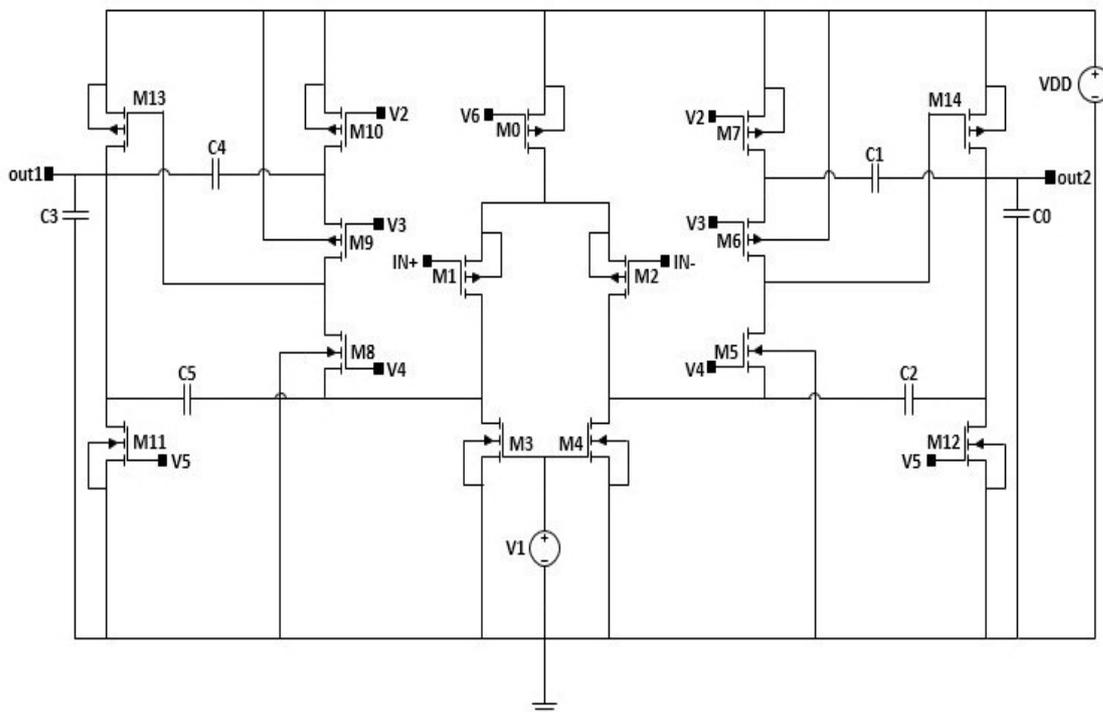


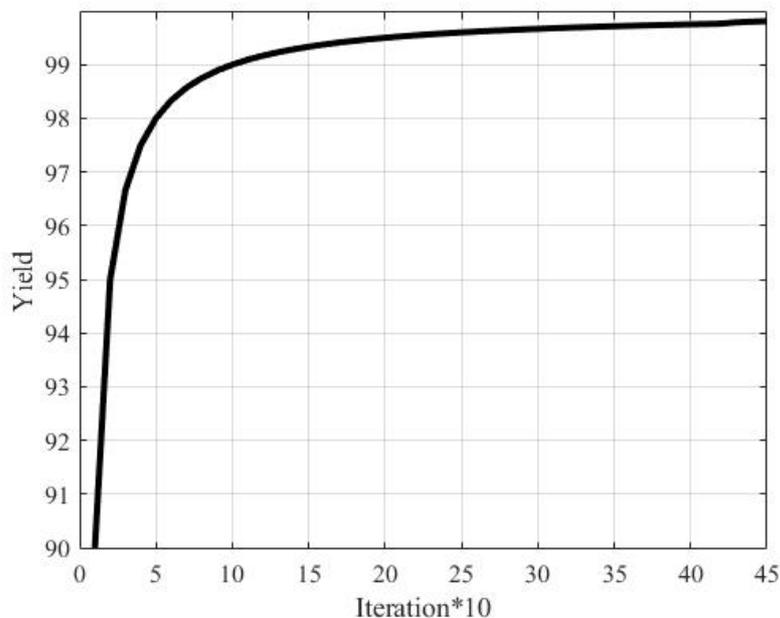
Figure 3. Two-stage fully differentially op-amp.

The CMOS technology used in circuit design is 180 nm and 1.8V power supply. The specifications are DC Gain  $\geq 65$  dB, unity gain bandwidth  $\geq 350$  MHz, Phase margin  $\geq 60^\circ$ , Output Swing  $\geq 1.9$  V, Power dissipation  $\leq 10$  mW, Slew rate  $\geq 390$  V / ms . In this method, design variables include the compensation capacitance, bias voltages, the number of parallel transistors, the transistor width, the transistor length and the threshold voltage. The transistor width and length are in the range of  $0.1\mu\text{m}$  to  $100\mu\text{m}$  and  $0.1\mu\text{m}$  to  $20\mu\text{m}$ , respectively. The range of the compensation capacitances changes from 0.1 pF to 10 pF.  $N_0 = 5$  is selected in stage one. Also, a value of 5 is selected for the parameter D in all simulations. The total budget of simulations is 10000.

To show the performance of the proposed method, simulations with 450 iterations for each method have been performed. The reason for choosing this number of iterations is that the yield variations after this value are almost constant. Also, another condition

considered in the first stage is that the solution with a minimum yield value of 90% will be selected to be sent to the next stage.

Figure 4, Table 1 and Table 2 show the result of the simulations and component's specification for the two-stage op-amp from one solution respectively. Using the proposed approach and according to Figure 4, the yield diagram starts at point 90% and ends at point 99.80%. The values of passive components, bias voltages, and size of transistors, for the two-stage op-amp from a solution are reported in Table 1. The simulation results of DC gain, unity-gain bandwidth, phase margin, power dissipation, output swing, and slew rate for the op-amp are given in Table 2. The results indicate that DC gain, unity-gain bandwidth, phase margin, power dissipation, output swing, and slew rate of the op-amp are 77.82 dB, 411.35 MHz, 62.5°, 7.68 mW, 3.28 V and 792.21 V/ $\mu$ s, respectively. By comparing the simulation results of Table 2 and the values considered as design constraints, it is observed that the results in addition to meeting the designer's specifications have excellent values for circuit design.



**Figure 4. The proposed yield method**

To compare the efficiency of the proposed method with a number of existing methods, we also tested the ORDE [15], the LHS [7] and the proposed algorithm in [24] on the circuit shown in Figure 3. As can be seen from the results of Table 3, the proposed algorithm is more accurate than the other two algorithms due to the use of MOCBA and the NSGA-III. However, the ORDE method has less computational time than the proposed approach due to its shorter computational steps. It is also observed that by using of the CA and MOCBA and eliminating unnecessary simulations for non-critical solutions, it has a higher efficiency than the LHS and OCBA method that using PSO. It also has a much lower computational load. Accordingly, it can be concluded that the proposed three-stage algorithm has acceptable accuracy and speed in comparison with the other methods.

**Table 1. Size of transistors, passive components, and bias voltages for the two-stage op-amp from one solution.**

Parameter	Value
$(W/L)_0$	5' 23.88 mm / 0.18 mm
$(W/L)_{1,2}$	49.78 mm / 0.18 mm
$(W/L)_{3,4}$	3' 22.71 mm / 0.18 mm
$(W/L)_{5,8}$	2' 28.92 mm / 0.18 mm
$(W/L)_{6,7,9,10}$	2' 80.39 mm / 0.18 mm
$(W/L)_{11,12}$	2' 23.88 mm / 0.36 mm
$(W/L)_{13,14}$	2' 80.39 mm / 0.36 mm
$C_{0,3}$	1 pF
$C_{1,2,4,5}$	1.05 pF
$V_1$	0.58 V
$V_2$	1.1 V
$V_3$	0.73 V
$V_4$	1.2 V
$V_5$	0.61 V
$V_6$	1 V

**Table 2. Specifications of the two-stage op-amp.**

No.	Specifications	Schematic-level simulation
1	DC gain (dB)	77.82
2	unity gain bandwidth (MHz)	411.35
3	Phase margin (dg)	62.5
4	power dissipation (mW)	7.68
5	Output Swing (V)	3.28
6	Slew rate ( $V/\mu s$ )	792.21

**Table 3. Yield results and Run time for three methods.**

Method	Best	Worst	Mean	Run time (h)
LHS	92.87	84.99	89.24	59
ORDE	99.35	97.46	98.14	22
OCBA+PSO	99.08	96.95	97.83	25
Proposed method	99.98	99.71	99.80	26

## 5. Conclusion

In this paper, a three-stage method has been presented having less computational effort and high accuracy compared to the MC based method. The proposed method used the CA to prevent repetition of over design. Also, by using the NSGA-III, the yield value has been enhanced. Another advantage is that it is generalizable so that it can be used for any high-performance analog circuit such as operational amplifiers or data

converters. The proposed approach has been tested on a two-stage op-amp. The results of the simulation substantiate our claim.

## References

- [1] I. Seghaier, M. H. Zaki, and S. Tahar, "Mating sensitivity analysis and statistical verification for efficient yield estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 2, pp. 294–307, 2020.
- [2] B. Ghanavati: High-Accurate Low-Voltage Analog CMOS Current Divider Modify by Neural Network and TLBO Algorithm. *Journal of Advances in Computer Research*, 8(1), 51-65.
- [3] M. Fakhfakh.: *Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design*. Springer International Publishing Switzerland 2015.
- [4] X-X. Liu, AA. Palma-Rodriguez, S. Rodriguez-Chavez, SX-D. Tan, E. Tlelo-Cuautle, Y. Cai.: Performance bound and yield analysis for analog circuits under process variations. In: 18th Asia and South Pacific Design Automation Conference (ASP-DAC). Singapore, 2013; 761–766.
- [5] W. Tian, X. Ling, R. Liu.: Novel methods for circuit worst-case tolerance analysis. *IEEE Transaction Circuits System I. Fundamental Theory and Applications* 1996; 43(4):272–278.
- [6] A. Canelas et al., "FUZY: A fuzzy c -means analog IC yield optimization using evolutionary-based algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 1, pp. 1–13, 2020.
- [7] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
- [8] A. Singhee and R. Rutenbar.: *Novel Algorithms for Fast Statistical Analysis of Scaled Circuits*. Berlin, Germany: Springer, 2009.
- [9] C.-H. Chen, J. Lin, E. Yucesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization'," *Discrete Event Dynamic Systems*, vol. 10, no. 3, pp. 251–270, 2000.
- [10] P. Glynn and S. Juneja, "A large deviations perspective on ordinal optimization'," 2004, vol. 1.
- [11] C-H. Chen. L. H. Lee.: *STOCHASTIC SIMULATION OPTIMIZATION An Optimal Computing Budget Allocation*. World Scientific Pub Co Inc; 1 edition 2010.
- [12] L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman, "Optimal computing budget allocation for multi-objective simulation models'," 2004, vol. 1.
- [13] L. H. Lee, E. P. Chew, S. Teng, and D. Goldsman, "Finding the nondominated pareto set for multi-objective simulation models'," *IIE Transactions*, vol. 42, no. 9, pp. 656–674, 2010.
- [14] S. Teng, L. H. Lee, and E. P. Chew, "Multi-objective ordinal optimization for simulation optimization problems'," *Automatica*, vol. 43, no. 11, pp. 1884–1895, 2007.
- [15] B. Liu, F. V. Fernandez, and G. E. Gielen.: Efficient and Accurate Statistical Analog Yield Optimization and Variation-Aware Circuit Sizing Based on Computational Intelligence Techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 6, June 2011.
- [16] Y. Liu, L. H. Lee, and E. P. Chew, "Multi-objective optimal computing budget allocation for multi-objective particle swarm optimisation with particle-dependent weights," *Int. J. Simul. Process Model.*, vol. 11, no. 3/4, p. 167, 2016.
- [17] B. Khamfroush, M. R. Akbari Jokar, K. Khamforoosh.: A Dual-Objective Nonlinear Model for Network Design with NSGA Algorithm. *Journal of Advances in Computer Research*, 10(3), 97-113.

- [18] A. Nodehi: Determining Cluster-Heads in Mobile Ad-Hoc Networks Using Multi-Objective Evolutionary based Algorithm. *Journal of Advances in Computer Research*, 9(3), 133-151.
- [19] K. Deb.: *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [20] C. M. Fonseca and P. J. Fleming.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. in *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufman, 1993, pp. 416–423
- [21] J. Horn, N. Nafplotis, and D. E. Goldberg.: A niched Pareto genetic algorithm for multiobjective optimization. in *Proceedings of the First IEEE Conference on Evolutionary Computation*, Z. Michalewicz, Ed. Piscataway, NJ: IEEE Press, 1994, pp. 82–87.
- [22] N. Srinivas and K. Deb.: Multiobjective function optimization using nondominated sorting genetic algorithms. *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Fall 1995.
- [23] K. Deb, Associate Member, IEEE, Amrit Pratap, Sameer Agarwal, and T. Meyarivan.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April 2002.
- [24] I. Guerra-Gomez, E. Tlelo-Cuautle, and L. G. de la Fraga, “OCBA in the yield optimization of analog integrated circuits by evolutionary algorithms,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015.

