Quantification of Design Difficulty of Analog Circuits Based on Volume of Effectual Design Space

Riku Anan, Akira Tsuchiya, Toshiyuki Inoue, Keiji Kishine
Department of Electronic Systems Engineering, The University of Shiga Prefecture
2500, Hassaka-cho, Hikone-shi, Shiga, Japan
tsuchiya.a@e.usp.ac.jp

Abstract— This paper proposes a method to quantify difficulty level of analog circuits. Analog circuits have many design parameters and performance measures, so it is difficult to set adequate target specifications as well as circuit design. However, there is less discussion about how to evaluate the design difficulty level quantitatively. We focus on the volume of design space where the performance meets the target specification. The proposed method calculates the volume of design space by using linear classifiers and slack variables. Numerical experiments show that the effective volume of design space can be used the measure of design difficulty level.

I. Introduction

Automated design of analog circuits has been a big challenge in integrated circuit design. Compared to digital circuits, analog circuits have wide variety of performance figures, and the design space is huge and high-dimensional. So, evaluating the performance and exploring the design space needs extremely high cost [1, 2]. Additionally, layout design and parasitics strongly affect the performance. Considering layout and parasitics makes the problem more complex. Thanks to progress of neural network, machine learning and AI technology, automated analog design also improved drastically [3].

However, it is not clear how to evaluate the performance requirement. Analog circuits have many performance figures, for example gain, power, bandwidth, noise, dynamic range, and so on. The relationship among these figures is complicated and intertwined. So, it is difficult to evaluate how difficult the performance requirement is. This is also related to process selection. To predict how high performance is achievable in a fabrication process, some basic parameters like transconductance g_m and its power efficiency $g_\mathrm{m}/I_\mathrm{D}$, cutoff frequency f_T are used. We can predict the performance of small circuits like single-stage amplifiers from these parameters, but it is not easy to predict larger circuits like opamp, VCO (Voltage-Controlled Oscillator) and so on. If we can know the difficulty of design at the early stage of design, it helps circuit design, target specification determination, and fabrication process selection.

The goal of this work is design-difficulty quantification of analog circuits. We propose to use a volume of effectual design space. Here, the word "effectual design space" means the region where the circuit performance meets the minimum requirement. If the effectual design space is large, many parameter sets can meet the minimum requirement, and it means the design difficulty is low. However, the design space of analog circuits is high-dimensional space. Exploring in such high-dimensional space to determine the effectual design space and calculating the volume of the effectual design space need huge computational cost. To solve this problem, we employ linear classifier and slack variable.

The contribution of this paper is a method to quantify the design difficulty of analog circuit. It is useful for early stage of circuit design, specification development, and fabrication process selection.

The rest of this paper is organized as follows. Section II explains the problem and the key idea. Section III describes the proposed method, and Section IV verifies by circuit simulation. Section V summarizes the discussion.

II. Effectual Design Space of Analog Circuit

This section describes "effectual design space" and its volume, which is the key to the proposed method.

A. Definition of Terms

First, we define some terms in this paper. The word *requirement* is a performance which the circuit under design has to satisfy. For example, gain, power consumption, bandwidth, phase margin, and so on. The circuit design has to meet all *requirements*. The work *design parameter* is a parameter which designers can tune. For example, channel length L, channel width W, and the value of passive components. Each *design parameter* has a range of values. The range is determined by design rules, area, and so on. In this paper, we assume that the circuit topology is fixed. All combinations of *design parameters* is called *design space*. Since all devices have 1 or more design parameters, the *design space* is high-dimensional space. For example, if a circuit has 10 transistors and each transistor can be tuned in L and W, the design space is 20-dimensional space. In design space, limited combination of design parameters can satisfy all requirements. We call this subspace where all requirements are satisfied as *effectual design space*.

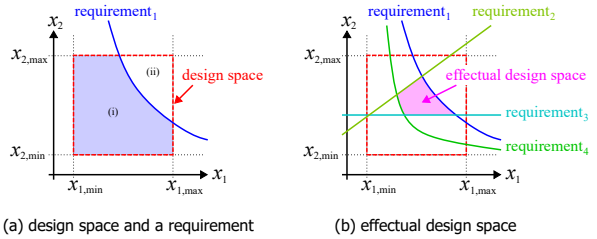

Fig. 1. Definition of design parameter, design space, requirement, and effectual design space.

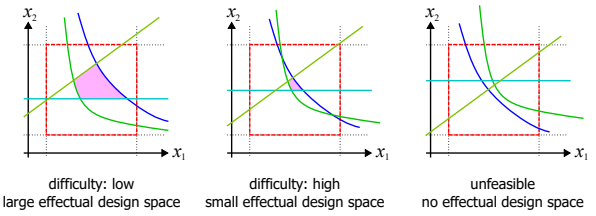

Fig. 2. Volume of effectual design space and design difficulty.

B. *Effectual Design Space and Its Volume*

A conceptual figure of effectual design space is show in Fig. 1. For simplicity, the design space is 2-dimensional space by two design parameters x_1 and x_2. In design space, a requirement corresponds to a border between meeting the requirement and not. In Fig. 1 (a), *requirement*$_1$ divides the design space into (i) and (ii). In many cases, analog circuits have multiple requirements. The design subspace where the circuit meets all requirements is a space enclosed by requirement borders as shown in Fig. 1 (b). We call this subspace as *effectual design space*.

The effectual design space is determined by requirements. As requirements change, the borders change, and the effectual design space changes as shown in Fig. 2. If the effectual design space is large, many sets of design parameters can meet the requirements, and the design difficulty is low. As the effectual design space becomes smaller, the design difficulty becomes higher. Depending on requirements, the effectual design space can disappear. It means that it is impossible to satisfy all requirements. In this case, designers have to change the requirement(s), circuit topology, or the fabrication process. We expect that the volume of effectual design space can be a measure of design difficulty. However, it is difficult to determine the effectual design space and calculate the volume especially in high-dimensional space. In the next section, we propose a method to calculate the volume of effectual design space.

III. Proposed Method for Volume Calculation of Effectual Design Space

The key idea of our design difficulty quantification is to use the volume of effectual design space as a measure of design difficulty. However, calculating the volume of arbitrary space is a class-#P problem. This section explains three techniques to obtain approximate solutions.

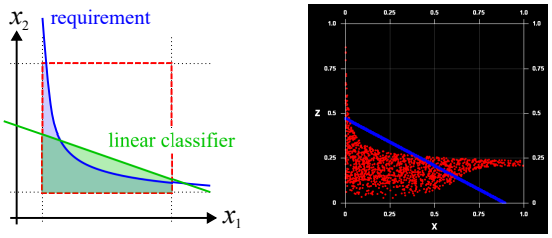

Fig. 3. Approximate border by a linear classifier.

A. *Approximate Effectual Design Space by Linear Classifier*

In n-dimensional space, a border determined by a requirement is a $(n-1)$-dimensional hypersurface. It is costly to even obtain one hypersurface, so it is difficult to obtain accurate effectual design space. To simplify the problem, we approximate a hyperplane instead of the hypersurface. Fig. 3 shows a concept in 2-dimensional space. The linear classifier is a linear border expressed by

$$a_0 + \sum_{i=1}^{n} (a_i \cdot x_i) \geq 0, \tag{1}$$

where a is a constant and x is a design parameter. n is the dimension of the design space. In Fig. 3 case, $n = 2$. We employ SVM (Support-Vector Machine) as linear classifier. As shown in Fig. 3 (a), actual border from a requirement is a curve (hypersurface). So, using a line (hyperplane) is rough approximation. Fig. 3 (b) is an example of a linear classifier in an amplifier design. X and Z are the normalized design parameters. Red dots are the samples that satisfy the requirement. As shown in Fig. 3, the envelope of the red dots is not linear. The blue line is the linear classifier. Since the border of the effectual design space is nonlinear, classification fails in some parts. The classification accuracy is lower than other complex classifiers such as nonlinear classifiers, but by using linear classifiers, approximate effectual design space becomes a convex hull. This enables effective evaluation when combined with the following two techniques.

B. *Volume Calculation by Markov-Chain Monte-Carlo Method*

For sampling in high-dimensional space, MCMC (Markov-Chain Monte Carlo method) is known as an efficient method [4–6]. We use random-walk sampling to evaluate the volume. According to Ref. [7], counting the solutions of 0-1 knapsack problem in polynomial time. For efficient random-walk sampling, all design parameters are discrete value expressed by a bit stream. As shown in Fig. 4, the range of each design parameter is divided by 2^{p_i} and expressed by p_i-bit parameter. In

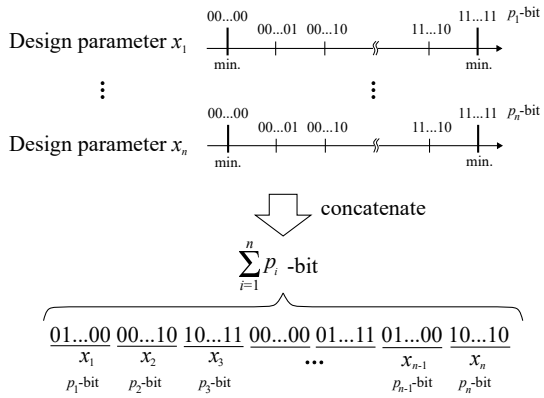

Fig. 4. Design parameter concatenation for random-walk sampling.

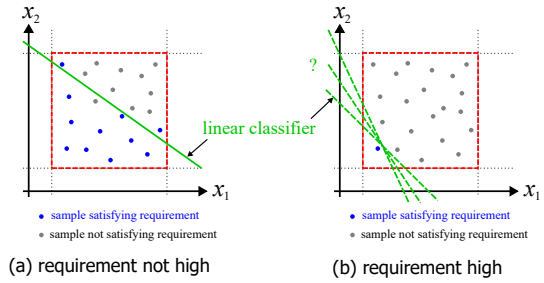

Fig. 5. Training data and linear classifier.

random-walk sampling, all design parameters are concatenated to one long bit-stream. For example, 10 design parameters are used and each of them are expressed by 5-bit, random-walk sampling is done on one $5 \times 10 = 50$-bit parameter.

C. Slack Variables

The proposed method is based on linear classifier. To obtain linear classifiers, we need training enough data because SVM is supervised learning. When the requirement is high, gathering training data becomes a problem. Fig. 5 shows simplified figure. When the training data has enough samples that satisfy and do not satisfy the requirement, the linear classifier is determined properly like Fig. 5 (a). If the requirement is high, the number of samples that satisfy the requirement becomes small. As shown in Fig. 5 (b), it is difficult to determine the linear classifier. However, it is difficult to increase the number of samples that satisfy the high requirement.

To solve this problem, we employ *slack variable*. Slack variable S is defined as

$$S = a_0 + \sum_{i=1}^{n} (a_i \cdot x_i), \tag{2}$$

that represents slide of the linear classifier. As shown in Fig. 6, when the slack variable S becomes larger, the linear classifier slides toward high-performance direction. By introducing

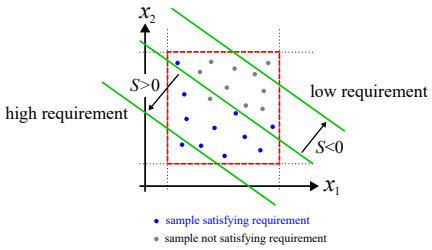

Fig. 6. Meaning of slack variable.

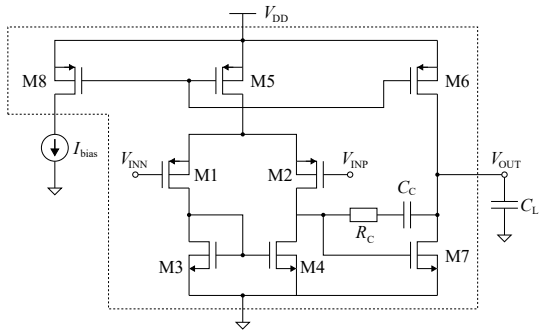

Fig. 7. Schematic of a two-stage operational amplifier.

slack variable, we can change the requirements without recreating the linear classifiers.

IV. Simulation Results

This section verifies the proposed method by circuit simulations. The system is implemented with Python. We use an open-source 130-nm CMOS PDK [8].

A. Circuit under Discussion

We use a two-stage operational amplifier (opamp) shown in Fig. 7. This circuit consists of 8 transistors, 1 resistor and 1 capacitor. Each transistor has two design parameters: channel length L and channel width W. Since the first stage is a differential pair, M1 and M2 are the same size transistor. Also, M3 and M4 are the same. M5, M6, and M8 are a current mirror, so the same channel length L is used for these transistors. Thus, there are 12 design parameters: W_1, W_3, W_5, W_6, W_7, W_8, L_1, L_3, L_5, L_7, R_C, and C_C. The supply voltage V_{DD}, the reference current source I_{bias} and the load capacitance C_L are fixed.

Table I shows the range of design parameters. #sample is the number of sampling points in each design parameter. In this experiment, each of 12 design parameters has 32 sampling points. So, the design space is 12-dimensional space, and the total number of sampling points is about 1.1×10^{18} (2^{60}). The requirements are gain, power, unity-gain frequency (UGF), output slew-rate, and input-referred noise.

TABLE I
DESIGN PARAMETERS OF THE OPAMP.

Parameter	min.	max.	#sample
Channel width W	1 µm	100 µm	32 (5 bit)
Channel length L	0.35 µm	1.0 µm	32 (5 bit)
Resistance R_C	0 Ω	2 kΩ	32 (5 bit)
Capacitance C_C	0.15 pF	5.0 pF	32 (5 bit)

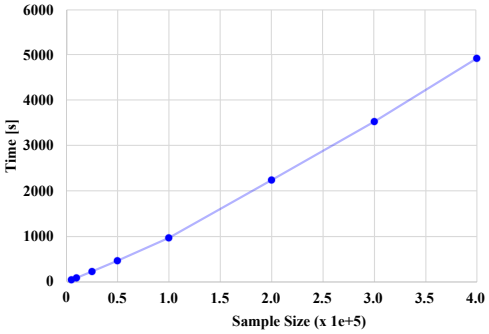

Fig. 8. Number of samples versus computational time for volume calculation.

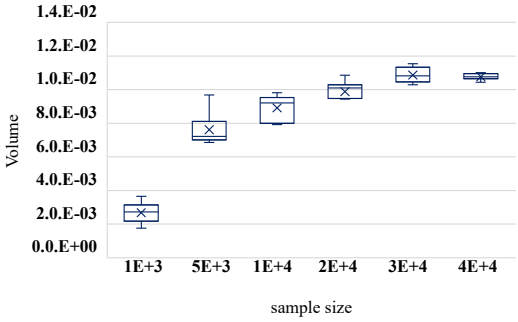

Fig. 9. Number of samples versus calculated volume.

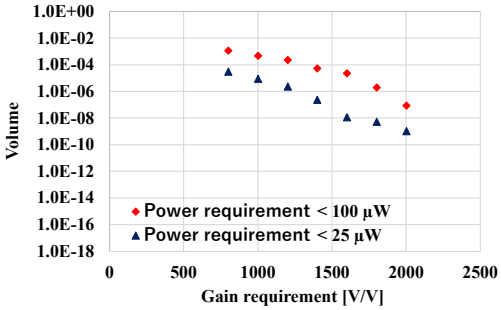

Fig. 10. Gain requirement versus the volume.

B. *Volume Calculation by MCMC*

First, we discuss the accuracy and computational time of volume calculation by MCMC method. We use random-walk sampling, so the estimated volume strongly depends on the sample size. Fig. 8 shows the sample size versus the time. The relationship is almost linear. Fig. 9 shows the relationship between the sample size and the calculated volume. Since MCMC method is a heuristic method, we tested 20 times at each sample size. The error-bar shows the distribution in the 20 trials. As the sample size increases, the calculated volume converges to 1.1×10^{-2}, and the error-bar becomes shorter. These results show that too small sample size causes estimation error, but accuracy improvement by increasing the sample size saturates. As shown in Fig. 8, the computational time is almost proportional to the sample size. In this case, 30,000 samplings can obtain enough accuracy in about 5 minutes on a 5.2-GHz-clock/16-core/32-GB-RAM PC.

C. *Relationship between Effectual Design Space and Performance Requirement*

We verify that the volume of the effectual design space can be used as a measure of design difficulty. When a requirement changes, the effectual design space changes and the volume changes. Here, slack variable is not used. When the requirements change, the linear classifier is recreated. Fig. 10 and Fig. 11 show the relationship between requirement and the volume. Please note that the y-axis is log-scale. In both cases of the gain requirement and the UGF requirement, the volume changes monotonically against the requirement change. When the power requirement is tightened from 100 µW to 25 µW, the volume reduces in all range of the gain/UGF requirement. These results show that the change of the volume is consistent with the change of requirements, thus the volume of effectual design space can be used as a measure of design difficulty.

D. *Maximum Performance Exploration by Slack Variable*

This subsection verifies that the slack variable if effective to calculate the volume with changing requirements. Also, we show the slack variable is useful to obtain the maximum performance that the circuit topology can achieve in the process.

As shown in Fig. 6, slack variable S indicates the amount of slide of linear classifier. It enables us to change requirement without recreating linear classifier. First, we evaluate the relationship between the slack variable and the circuit performance. After creating linear classifiers with a set of requirements, the borders are shifted according to the slack variable. As the slack variable increases, the effectual design space becomes small. Samples inside the narrowed effectual design space meet tighter requirements and achieve higher performance. Fig. 12 shows the distribution of the gain inside the effectual design space. As the slack variable increases, the distribution shifts toward higher gain. This result shows that sliding the borders by the slack variable can catch appropriate

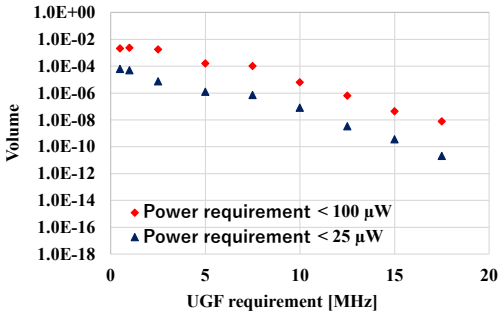

Fig. 11. UGF requirement versus the volume.

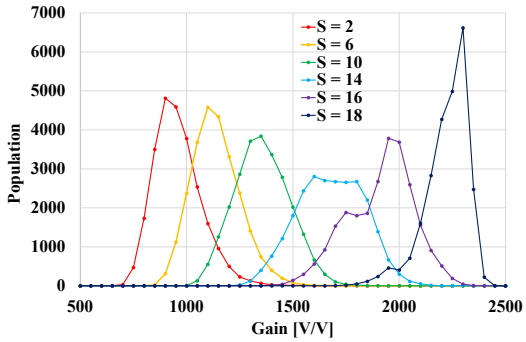

Fig. 12. Change of performance distribution by slack variable.

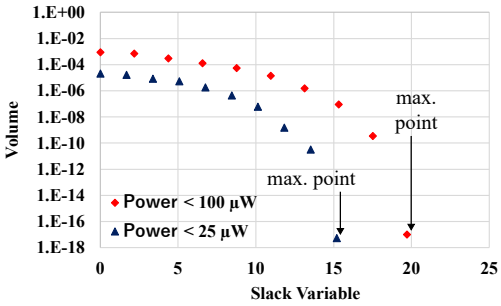

Fig. 13. Gain requirement versus the volume.

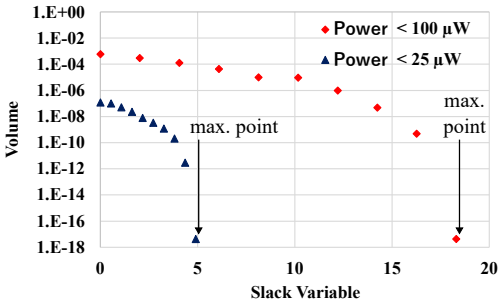

Fig. 14. UGF requirement versus the volume.

subspace corresponding with the effectual design space.

Then, we can estimate the point where the effectual design space disappears. This point means that the maximum achievable performance of the circuit. Without the slack variable, it is difficult to find the point because it is difficult to catch enough number of samples around this maximum performance point.

Fig. 13 is the results when a slack variable is applied to the gain requirement. At $S = 15$ (power requirement $< 25\,\mu W$) and $S = 20$ (power requirement $< 100\,\mu W$, the estimated volume becomes 1×10^{-17} at these points, the effectual design space is assumed to be disappeared. Fig. 14 is the case of UGF. Compared to Fig. 13, the effectual design space disappears at $S = 5$ when the power requirement is $< 25\,\mu W$. From these results, the maximum UGF is strongly affected by the power requirement.

To verify the results obtained by the slack variable, we compare the results by a multi-objective optimization developed for analog circuit design using NSGA-II (Non-dominated Sorting Genetic Algorithms II) [9]. Trade-off curves obtained by the proposed method and Ref. [9] are shown in Figs. 15–18. In all trade-off curves among the gain, the power, the slew-rate, and the input-referred noise, the results by the proposed method and NSGA-II show good matching. This means that the proposed method using slack variable captures the performance limitation without re-training of linear classifiers.

From discussion above, the slack variable is also useful to evaluate how one performance requirement affects the limitation of other performance measures.

V. Conclusion

This paper proposed a method to evaluate the volume of effectual design space of analog circuits. The volume of effectual design space correlates design difficulty, but it is difficult to obtain the volume because the effectual design space is a subspace in high-dimensional design space. We employed three techniques for efficient volume calculation: (1) linear classifier, (2) MCMC, and (3) slack variable. The proposed method achieved quantification of design difficulty in practical computational cost. By using slack variables, we can find the maximum performance point of the circuit under design. Then, we can evaluate quantitatively how one performance requirement affects the maximum point in other performance measures. The proposed method is useful at early stage of circuit design, target specification determination, process selection.

References

[1] L. Zhang, S. Raju, A. James, R. Dutta, G. Fournier, D. Lancry, K. C. T. Chuan, V. R. Chandrasekhar, and C. S.

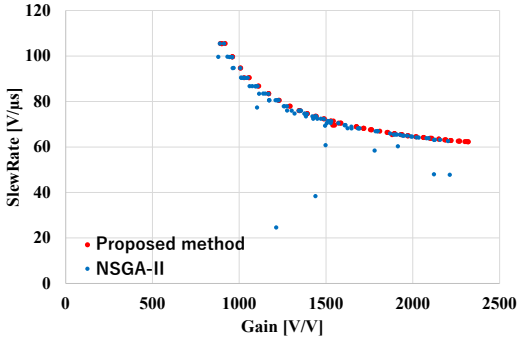

Fig. 15. Trade-off between the gain and the slew rate.

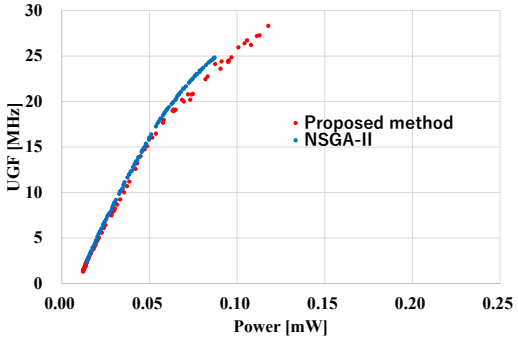

Fig. 17. Trade-off between the power and the UGF.

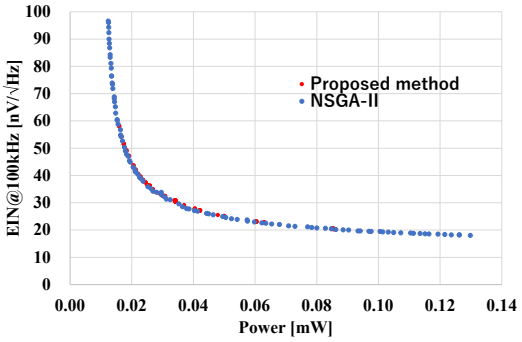

Fig. 16. Trade-off between the power and the input-referred noise.

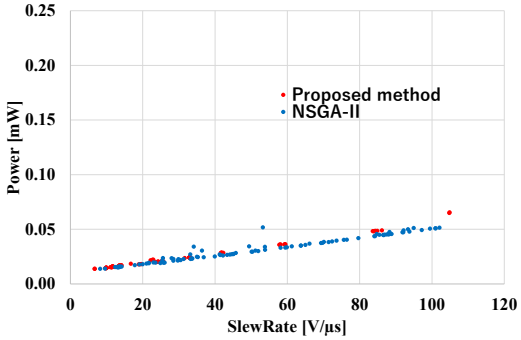

Fig. 18. Trade-off between the slew rate and the power.

Foo, "Bayesian deep active learning for analog circuit performance classification," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 3018–3022.

[2] R. Lyu, Y. Meng, A. Zhao, Z. Bi, K. Zhu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "A study on exploring and exploiting the high-dimensional design space for analog circuit design automation : (invited paper)," in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2024, pp. 671–678.

[3] G. Linán-Cembrano, N. Lourenço, N. Horta, and J. M. de la Rosa, "Design automation of analog and mixed-signal circuits using neural networks—a tutorial brief," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 3, pp. 1677–1682, 2024.

[4] J. S. Speagle, "A conceptual introduction to Markov Chain Monte Carlo methods," 2020. [Online]. Available: https://arxiv.org/abs/1909.12313

[5] N. T. Hunt-Smith, W. Melnitchouk, F. Ringer, N. Sato, A. W. Thomas, and M. J. White, "Accelerating Markov Chain Monte Carlo sampling with diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2309.01454

[6] I. Beichl and F. Sullivan, "The metropolis algorithm," *Computing in Science & Engineering*, vol. 2, no. 1, pp. 65–69, 2000.

[7] B. Morris and A. Sinclair, "Random walks on truncated cubes and sampling 0-1 knapsack solutions," in *40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 230–240.

[8] SkyWater PDK authors, "SkyWater SKY130 PDK," https://github.com/google/skywater-pdk.

[9] A. Sasikumar and R. Muthaiah, "Operational amplifier circuit sizing based on NSGA-II and particle swarm optimization," in *2017 International Conference on Networks & Advances in Computational Technologies (NetACT)*, 2017, pp. 64–68.