Circuit-centric Genetic Algorithm (CGA) for Analog and Radio-Frequency Circuit Optimization

Yeonjun Lee[1]†, Mingi Kwon[1]†, Ickhyun Song[1]*

[1]Department of Electronic Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, 04763, Seoul, Republic of Korea.

*Corresponding author(s). E-mail(s): isong@hanyang.ac.kr;
Contributing authors: yeonjun3333@gmail.com;
mingi001025@gmail.com;
†These authors contributed equally to this work.

Abstract

This paper presents an automated method for optimizing parameters in analog/high-frequency circuits, aiming to maximize performance parameters of a radio-frequency (RF) receiver. The design target includes a reduction of power consumption and noise figure and an increase in conversion gain. This study investigates the use of an artificial algorithm for the optimization of a receiver, illustrating how to fulfill the performance parameters with diverse circuit parameters. To overcome issues observed in the traditional Genetic Algorithm (GA), the concept of the Circuit-centric Genetic Algorithm (CGA) is proposed as a viable approach. The new method adopts an inference process that is simpler and computationally more efficient than the existing deep learning models. In addition, CGA offers significant advantages over manual design of finding optimal points and the conventional GA, mitigating the designer's workload while searching for superior optimum points.

Keywords: artificial intelligence (AI), automation and optimization, modified genetic algorithm, low-noise amplifier (LNA)· mixer, radio-frequency (RF) receiver

1 Introduction

Artificial intelligence (AI) has made significant advancements in various fields and it has been widely applied across various domains. Continuous semiconductor scaling enhances hardware performance, while improvements in the quality of vast input data enable AI to surpass human capabilities in numerous areas. In this regard, the circuit design process is also under consideration for AI applications, given its need to deal with complex interactions between mathematical calculations and various components. The optimization algorithms proposed so far, however, are computationally intensive and resource-hungry, resulting in prolonged design times, reduced design process efficiency, and increased economic costs due to excessive power consumption. [1]

In the literature, there have been several approaches to circuit design and optimization. Many researchers have focused on circuit creation based on expert knowledge [2], [3]. However, making a good technical decision within a vast design space that encompasses all possible circuits is challenging. Whereas several nature-inspired algorithms such as PSO (particle swarm optimization) and ACO (ant colony optimization) have shown promise in solving problems, however, they still face difficulties in terms of accuracy and computation time for multivariate and multimodal problems.[4]

Genetic Algorithms (GAs) are recognized for their efficiency in conducting comprehensive global searches. They excel in maintaining a population of diverse solutions, enabling the exploration of extensive search spaces. These attributes render GAs particularly well-suited for optimizing circuit designs, a domain characterized by intricate, multimodal, and non-convex fitness landscapes that challenge the identification of global optima. Furthermore, GAs exhibit a high degree of adaptability, making them versatile across various problem domains and solution representations. Their applicability extends seamlessly to both discrete and continuous optimization problems. GAs empower users to finely tailor their choice of encoding and genetic operators to precisely align with the specific nuances of the given problem. Additionally, the intrinsic ability of GAs to uphold genetic diversity within their populations is invaluable. Mechanisms like mutation and recombination facilitate this diversity maintenance [5]. Such inherent diversity plays a pivotal role in guiding GAs away from local optima, allowing for a more thorough exploration of the search space. GAs offer a powerful and adaptable approach for circuit design optimization, particularly in scenarios with complex fitness landscapes, setting them apart from PSO and ACO.[6] practice.

High-frequency (RF) circuits are composed of various parameters, including noise, voltage gain, input-output matching, stability, power consumption, and noise figure, all of which must be satisfied. RF circuit design is characterized by heightened sensitivity to parameter variations, where even slight changes can significantly impact performance. This heightened sensitivity extends to both analog and high-frequency characteristics, demanding a comprehensive consideration of parameters in both domains. The intricate interplay of analog and high-frequency traits necessitates meticulous attention to factors such as impedance matching, signal integrity, and power handling. Furthermore, the influence of noise becomes substantially more pronounced in RF circuits. The impact of noise on signal quality and system performance is magnified, requiring sophisticated noise mitigation strategies and careful design considerations. In the delicate balance of optimizing parameters, RF circuit designers must navigate the challenges posed by the increased susceptibility to variations and the pervasive influence of noise, making the design process inherently more intricate and demanding.

RF circuits and transceivers involve numerous trade-offs, especially as circuits become more complex [7]. Quantitative analysis of the impact of parameters on the final performance and design variables becomes limited as circuits become more complex, and manually finding optimal points for each parameter as analog circuit design parameters increase heavily depends on the designer's experience and heuristic approaches. Trade-off relationships between specific performance metrics and others can complicate the optimization process. For example, lowering the noise of a front-end amplifier may require higher power consumption or sacrifice linearity [7].

To address these challenges, this paper introduces a modified GA termed circuit-centric genetic algorithm (CGA) to conduct receiver (Rx) optimization, offering a simple yet effective approach. An algorithm suitable for transistor-level circuit design transfer was developed by modifying the mutation method and generational transfer method in traditional GA. Research is currently underway to enhance the efficiency of circuit design engineers using Python-based automation to control Cadence tools and other design automation techniques [8]. This study was also written in Python and utilized the free commercial tool LTspice to automatically optimize circuit performance. The optimization results were evaluated using the figure of merit (FoM), which includes key performance metrics for the receiver, such as gain, power consumption, and noise figure.

The proposed circuit design in this study demonstrated a 30% improvement in FoM compared to existing circuits, validating the efficiency of circuit design using genetic algorithms and showcasing its applicability to various analog circuit designs.

2 Receiver Design and Figure of Merit

One of the key modules in general RF systems is an RF receiver (Rx), which essentially performs detection, amplification, and frequency translation of the incoming RF inputs, resulting in an intermediate frequency (IF) signal for subsequent data processing [9]. The block diagram of a typical Rx is shown in Fig. 1. An antenna is used to receive wireless signals at a specific frequency. It captures radio waves through the air, converts them into electrical signals, and transmits them to the receiver. As the first gain stage of an Rx an LNA is used to amplify a very weak signal from the antenna with minimal addition of noise to the input. An ideal LNA should provide a high gain and a low noise figure (NF) [10]. After the LNA stage, a down-conversion mixer conducts frequency-mixing operation, using a local oscillator (LO) signal from a voltage-controlled oscillator (VCO) and generating a voltage output at intermediate frequency (IF). LO is a local oscillator commonly used in receivers. It generates a specific frequency to select the desired frequency without interference from other signals. The VCO, responsive to changes in voltage, allows for dynamic control of the LO's frequency. Since a mixer may generate relatively high noise, it usually is placed after the LNA for noise minimization of the entire Rx chain [7]. Additionally, placing an IF Filter after the down conversion mixer is crucial to isolate the desired signal from multiple frequency components and enhance the receiver's selectivity for effective signal extraction [11].

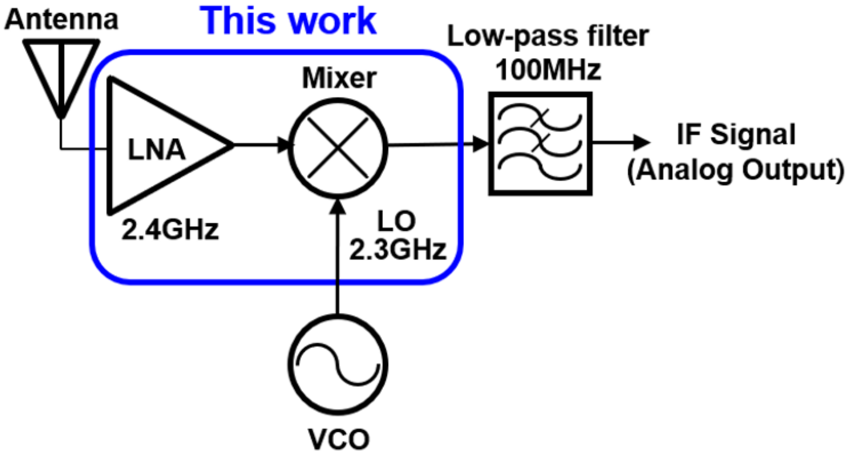

Fig. 1 System configuration of an RF receiver (Rx) in which an antenna, an LNA, a down-conversion mixer, a voltage-controlled oscillator (VCO), and a low-pass filter are included. The frequencies of the input, the local oscillator (LO) signal, and the output are 2.4 GHz, 2.4 GHz, and 100 MHz, respectively.

The overall configuration of the Rx is based on the example in Razavi's 'RF Microelectronics 2nd edition' [7] and the combination of the LNA and the mixer will be explored throughout this study designed for operation in the 2.4GHz band. The circuit includes an LNA for noise reduction and increased gain, with a down-conversion mixer connected at the back of the LNA. The entire receiver is configured for down conversion to a 100MHz IF frequency using a 2.3GHz Local Oscillator. The power supply voltage is set at 1.2V. Single Balanced Mixer (SBM) is selected for its simplicity and versatile performance, making it suitable for general-purpose use compared to the more complex Double Balanced Mixer (DBM). The LNA and SBM are designed using 0.18 µm CMOS Technology [11]. The circuit was designed and simulated using built-in components in LTspice, and LTspice was controlled using Python for automation.

The designed circuit as shown in Fig. 2. represents the block before C3 as an LNA, followed by a mixer in the next part. PMOS M1 serves as both a load and an amplifying device, yielding a lower noise figure than if the load is passive. Current source I1 defines the bias of M1 and M2, and C1 creates an ac ground [7]. NMOS M2 is connected to act as a buffer for the LNA. Setting R1 with a large value facilitates the establishment of an appropriate DC level at the gates of M1 and M3, thereby enabling a higher voltage gain. The mixer is designed by increasing the PMOS sizes

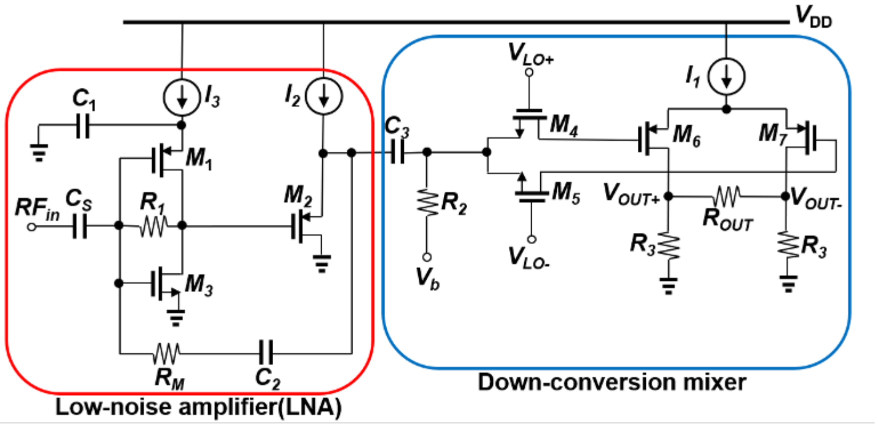

Fig. 2 Full schematic of RF receiver including an LNA and a single-balanced down -conversion mixer.

of M6 and M7 to minimize flicker's noise. Vb defines the gate bias voltage for the differential pair, maintaining a level of 0.2 V [7].

The LNA plays a role in amplifying the input signal and minimizing noise, while the mixer performs downconversion for data processing at intermediate frequencies. Strongly coupling these two components enhances the overall circuit performance, and it allows finding optimal solutions in cases where certain trade-offs occur. Therefore, optimizing the LNA and mixer as a unified block can improve the efficiency and performance of the entire receiver system.

Although there are many performance factors in LNAs, we concentrate on only three important parameters such as voltage gain, noise figure and power dissipation [12].

Considering that in a cascaded circuit, the noise figure is mainly affected by the value of the first block, the noise figure for the entire receiver was approximated by solely measuring the noise figure of the LNA. (The LNA gain obtained from the actual results is greater than 10dB, so taking the mixer's noise figure into account did not significantly change the figures.) The overall noise figure of the cascaded circuit was calculated by referencing the Friis equation as follows:

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} + \dots + \frac{F_{n-1} - 1}{G_1 G_2 G_3 \dots G_{n-1}} \tag{1}$$

For the measurement of downconversion gain, the large signal model was applied to calculate the maximum differential output voltage of the receiver's output relative to the maximum RF voltage input,set at an amplitude of 0.3. Power consumption was determined by averaging the power consumed by all circuit components during the entire 50ns operational period.

The entire algorithm progressed through simulations aimed at maximizing the FoM by calculating three parameters. However, it was necessary to prevent the FoM from excessively increasing in the undesired direction of the circuit's specifications. For instance, where the noise figure exceeds 5 but the down conversion gain significantly compensates for it, the value of the noise figure was replaced with 10,000 to hinder an anomalous increase in the FoM. This method was taken to design the circuit within the desired specification range.

$$FoM = \frac{Conversiongain[dB]}{Noisefigure[dB] \cdot Powerconsumption[W]} \tag{2}$$

3 Genetic Algorithm Adaptation for Analog and Radio-frequency Circuit

3.1 Genetic Algorithm

A Genetic Algorithm (GA) is a heuristic search algorithm used to solve optimization problems. It is based on the fundamental concept that ”all organisms survive by adapting to various environments,” which was proposed by Charles Darwin’s theory of natural selection. Genetic Algorithms were first introduced by John Holland in the 1960s and have since been applied as a paradigm of natural evolution [13]. They are based on concepts such as genes, mutations, natural selection, and crossover.

Utilizing these concepts, GAs form a new candidate solution set and iteratively find the optimal solution over multiple generations. The candidate solutions within each generation’s solution set are referred to as chromosomes or individuals [14].

3.2 Advantages of Genetic Algorithm

Genetic Algorithms do not require constraints such as continuity or differentiability of the objective function, unlike conventional methods. This is because designers determine the fitness function in a way that promotes an increase in the desired direction, and the algorithm solely computes this function’s values for optimization. Therefore, Genetic Algorithms offer the advantage of being relatively easy to apply to real-world systems without the need for additional information beyond fitness evaluation.

3.3 Basic Principles of Genetic Algorithm

Genetic Algorithms are based on concepts such as genes, natural selection, crossover, and mutation. Natural selection is the process of determining which individuals within the current generation will be passed on to the next generation. Typically, individuals with higher fitness are selected. [16] The fitness function measures how good each individual’s solution is, and in this study, FoM was used. FoM is calculated as the ratio of Gain to the product of Power Consumption and Noise Figure, where higher FoM values represent superior performance.

Crossover, on the other hand, involves exchanging information between selected parent individuals to create new offspring individuals. Crossover can be performed in various ways, depending on the characteristics of the Genetic Algorithm. [15]

Mutation introduces diversity by randomly altering an individual’s genes. It occasionally introduces new information and makes the optimization process more exploratory.

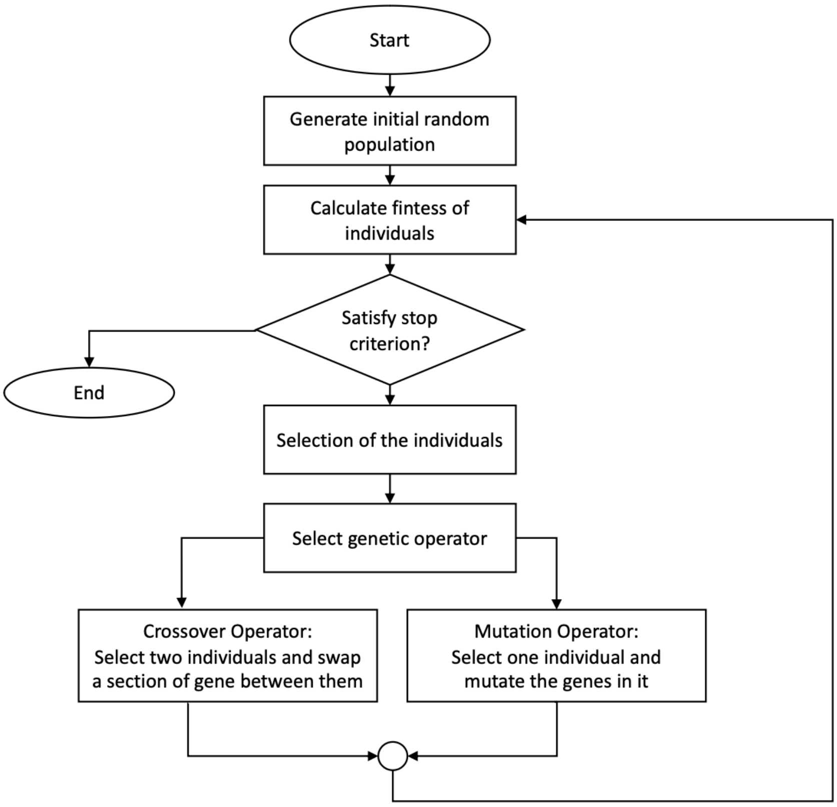
Start
Generate initial random population
Calculate fintess of individuals
Satisfy stop criterion?
End
Selection of the individuals
Select genetic operator
Crossover Operator: Select two individuals and swap a section of gene between them
Mutation Operator: Select one individual and mutate the genes in it

Fig. 3 Genetic Algorithm flowchart[15]

In this paper, the algorithm designed is more suitable for circuit design automation by omitting the crossover process and enhancing the mutation process. First, the drawbacks of using the unaltered Genetic Algorithm are discussed, and then the improved algorithm is described to highlight its potential for circuit enhancements.

3.4 The Challenges of Applying Genetic Algorithms in Circuit Design

The graph above represents the FoM graph computed using the Genetic Algorithm (GA), which effectively illustrates why GA may not be suitable for circuit design.

Optimization was carried out over a total of five generations. In the initial generation, 81 individuals were randomly generated. The FoM values obtained were then sorted in descending order, and the top 1/3 of individuals, which corresponds to the top 27 individuals, were selected. Subsequently, a crossover operation was applied to

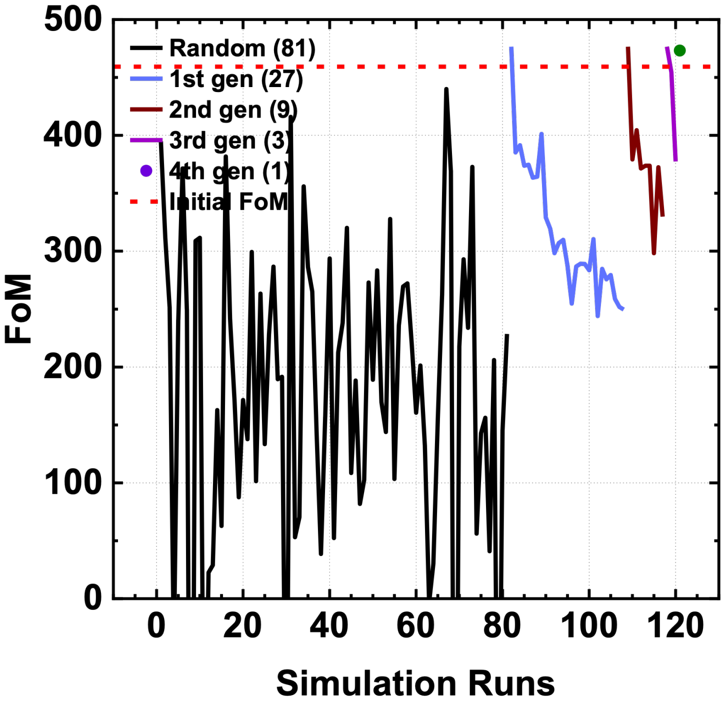

Fig. 4 FoM of circuit designed with original GA algorithm

these selected individuals. From the graph, it is evident that in the first generation, the individual with the highest FoM is the 67th individual, with a FoM of 439.92.

One randomly selected component among the twenty components of this individual was specified. For this specified component, a new random value was generated within the predefined range for that component. FoM was recalculated for the individual after this mutation operation. In the case of the 67th individual, after the mutation operation, its FoM increased to 476.58.

The graph above highlights the inherent issues with applying the genetic algorithm directly to this research. It becomes evident that Post consistently does not exhibit higher FoM values compared to Prior. The average FoM of the top 27 individuals in the initial generation is 315.94, but the average FoM for the second generation, which undergoes mutation operations on these initial individuals, only slightly increases to 316.28, indicating insufficient performance improvement.

Furthermore, the average FoM of the top 9 individuals in the second generation is 384.45, yet the average FoM of the third generation, which comprises the results of the mutation operations on these top-performing individuals, decreases to 375.56. This contradicts the fundamental principle of genetic algorithms, where over generations, only superior individuals should survive and evolve.

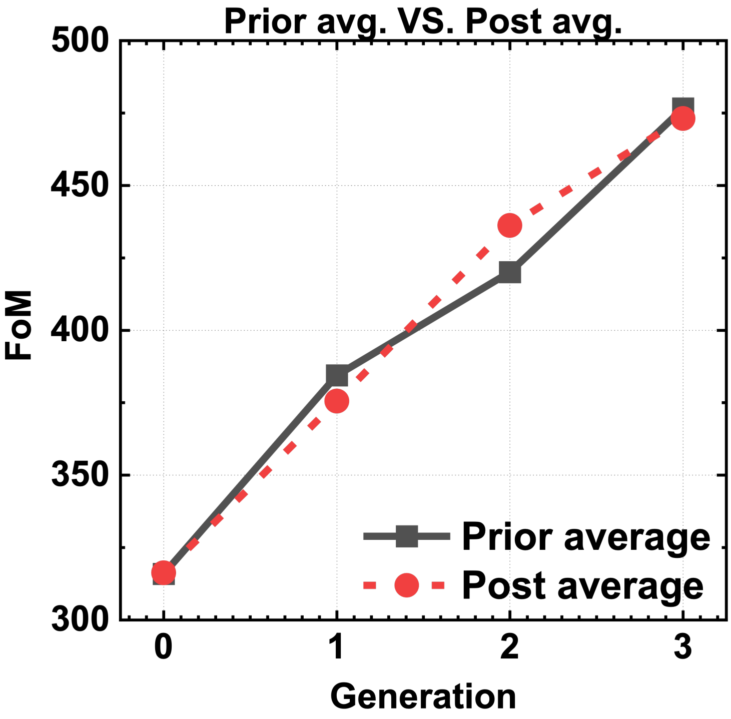

Fig. 5 System configuration of an RF receiver (Rx) in which an antenna, an LNA, a down-conversion mixer, a voltage-controlled oscillator (VCO), and a low-pass filter are included. The frequencies of the input, the local oscillator (LO) signal, and the output are 2.4 GHz, 2.4 GHz, and 100 MHz, respectively.

This problem arises because individuals with initially high FoM values tend to lose their distinctive characteristics through information exchange with other individuals during the crossover process. Therefore, the mutation operations via genetic algorithms do not effectively facilitate the clear learning and improvement of FoM.

3.5 CGA : Circuit-Centric Genetic Algorithm

This paper addresses the problem of finding optimal component values for LNA (Low Noise Amplifier) and Mixer circuits. Genetic Algorithms (GAs) typically generate offspring generations through the crossover of superior individuals. However, when applying GA to this research, it encountered issues of randomness, such as a significant drop in the overall FoM due to the crossover of circuits where Noise Figure values dominated, despite the component values for Gain being superior.

In response to these challenges, a new approach called Circuit-centric Genetic Algorithm (CGA) was devised and adopted.

Algorithm 1 Genetic Algorithm for Electronic Circuit Design

```
Initialize parameters:
        $n_{\text{gen}}$, pop_size, target_FOM, max_simulations
Initialize best_FOM, best_individual, simulations_run to 0
Create an initial population of individuals:
        Each individual represents a set of circuit parameters
        Randomly generate these parameters within specified ranges
while simulations_run < max_simulations and best_FOM < target_FOM do
    Evaluate each individual in the population:
    for all individuals do
        Circuit modification
        Run LTspice simulations to obtain:
                Gain
                Power consumption
                Noise figure
        Calculate the Figure of Merit (FOM)
    end for
    Sort the evaluated population based on FOM (highest to lowest)
    Select the best-performing individual
    for all selected individuals do
        Mutatation
        Run LTspice
        If the FOM improves, update the individual;
    end for
    Check termination conditions:
    if best_FOM $\geq$ target_FOM or simulations_run $\geq$ max_simulations then
        Exit loop
    end if
    Update the population with the newly created individuals
end while
```

3.6 CGA operation

The provided pseudo code outlines the modified CGA algorithm. In the initial generation, 30 individuals are randomly generated. Among these generated individuals, the one with the best performance, meaning the highest FoM, is selected. For this chosen individual, the mutation algorithm is applied a total of 20 times. The reason for performing this operation 20 times is to apply a mutation operation once for each component variable.

From these newly modified twenty individuals, the one that exhibits the highest FoM is reassigned. The same process is then repeated for this re-assigned individual. By employing this adapted algorithm, the system is designed to continuously improve the component values of individuals that demonstrate superior performance.

3.7 CGA's distinct feature compared to GA

The key distinctions between CGA and the traditional GA lie in the absence of a crossover process in CGA and the continuous application of mutation operations exclusively to a single individual. In traditional GA optimization, a certain proportion of individuals from the current generation typically engage in crossover. However, in CGA, instead of crossover a fraction of individuals from the generation, the chosen approach involves performing mutation operations for the best FoM individual for each component variable, and this is repeated twenty times.

This fundamental difference between GA and CGA addresses the issue at hand. The improved algorithm boldly focuses on a single individual with the highest FoM, bypassing others, and applies mutation operations for all twenty component variables. This significantly increases the possibility of improving FoM. While it might be conceivable to perform mutation operations for multiple individuals with top FoM values, the decision to focus on a single top-performing individual is made to preserve the simplicity inherent in genetic algorithms and reduce computational complexity.

This approach ensures that FoM can continuously improve over generations by reintroducing randomness while preserving parameters that maintain a favorable FoM, which sets it apart from the traditional GA.

4 Optimization Results and Discussion

4.1 Results

When examining the FoM graphs, it becomes evident that unlike traditional GA, which did not consistently show an upward trend with each generation, CGA displays a clear upward trajectory. This increase is not merely reliant on the randomness of the initial generation but is confirmed to be a result of the principles of the optimization algorithm. Observing the graphs of individual components contributing to FoM, it is notable that Gain increases, while power consumption and noise figure decrease. This illustrates that FoM improvement is not solely driven by a single parameter but involves the simultaneous exploration of optimal values for all three parameters.

Notably, the significant improvement in the final FoM is predominantly driven by the lower noise figure, contributing to approximately a 43.5

This flexibility in CGA allows circuit designers to easily tailor the FoM formula or set threshold values for specific parameters based on their priority for improvement. This enables the straightforward adjustment of the FoM to focus on specific aspects of circuit performance, depending on the designer's objectives.

4.2 Discussion

The algorithm adopted in this paper, known as CGA, has several advantages over traditional algorithms like GA when it comes to optimizing analog circuit designs. It holds multiple merits and challenges:

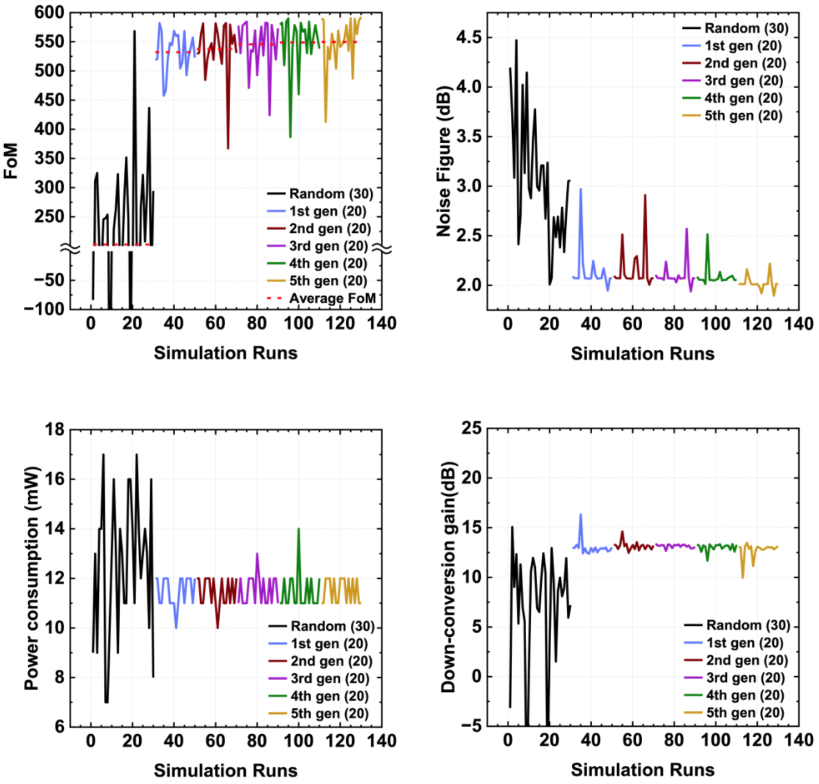

Fig. 6 FoM steadily rising with its parameters gain, power consumption and noise figure

4.2.1 Merits

1. Less Computation: Unlike complex deep learning algorithms that require extensive training time, our algorithm utilizes minimal computation to achieve the desired specifications in circuit designs.
2. Randomness & Low Power: Our algorithm introduces randomness to overcome issues such as overfitting and erroneous optimization points, similar to how deep learning models address these problems by incorporating momentum factors and using methods like MBGD [17]. However, deep learning techniques often involve additional parameters and higher power consumption. In contrast, our approach, which relies on randomness, offers simplicity and freedom from such challenges.
3. Versatility & Programmability: The algorithm's simplicity allows it to be applied to various circuit blocks beyond RF circuits. Designers can easily prevent circuit components from deviating from their intended specifications, promoting flexible and tailored circuit designs. Unlike current deep learning techniques, which can become challenging when adapting to changes in input data or component ranges, our algorithm exhibits high re-programmability.

Table 1 Optimal circuit component variables drawn by the Circuit-centric Genetic Algorithm

Parameter	Initial Value	Optimal Value	Range	Step size
R1 (Ω)	5,000	5,146	3,000 – 6,000	1
R2 (Ω)	2,000	3,219	1,500 – 4,000	1
R3 (Ω)	200	897	1 – 1,000	1
Rout (Ω)	100,000	8,078	1,000 – 100,000	1
Rm (Ω)	450	998	100 – 1,000	1
C1 (pF)	2	2.92	1.5 – 3	0.01
C2 (pF)	1	1.05	0.1 – 5	0.01
C3 (pF)	0.5	0.86	0.1 – 5	0.01
Vb (V)	0.2	0.11	0.1 – 0.4	0.001
I1 (mA)	5	3.69	3 – 7	0.01
I2 (mA)	2	3.33	1 – 5	0.01
I3 (mA)	2	3.56	1 – 5	0.01
VLO offset (V)	0.5	0.66	0.3 – 0.7	0.01
VLO Amp (V)	0.4	0.65	0.2 – 1.0	0.01
M1 width (µm)	144	101	90 – 180	1
M2 width (µm)	14.4	13.57	9 – 18	0.01
M3 width (µm)	5.4	4.04	1.8 – 9	0.01
M4 width (µm)	0.9	1.63	0.54 – 1.8	0.01
M5 width (µm)	0.9	1.63	0.54 – 1.8	0.01
M6 width (µm)	15	16.81	9 – 18	0.01
M7 width (µm)	15	16.81	9 – 18	0.01

Table 2 Gain, power consumption and noise figure of optimal circuit

Parameter	Initial Value	Optimal Value
Gain (dB)	16.35	13.13
Power (W)	0.01	0.011
Noise figure (dB)	3.56	2.01
FoM	459.30	592.67

Table 3 Fixed parameters list and its values

Parameter Name	Value
VDD (V)	1.2
RFin amp (V)	0.3
CMOS Length (μm)	0.18

4.2.2 Challenges

1. Dependency on Initial Values: The algorithm relies on manual configuration of the ranges and initial values of all circuit components. The circuit's final optimal point can vary significantly depending on the settings, emphasizing the importance of the designer's knowledge and understanding of component ranges.

2. Dependency on Randomness: The algorithm exhibits a heuristic and prominently random nature. If the initial generation's randomly selected individuals with the best FoM are limited, the algorithm may not achieve the desired performance,

even after several generations. Increasing the number of individuals in the initial generation can address this issue, but it involves a trade-off with computational resources and training time.
3. Non-Convergence: While many deep learning algorithms aim for convergence, ensuring stable algorithmic outcomes, our algorithm, with its inherently random nature, may not converge to a fixed optimal point over successive generations. While it can find circuit designs that meet the desired specifications over time, it may not reach a perfect solution.

5 Summary

Considering the merits and challenges discussed, the algorithm used in this research is argued to offer significant convenience from the perspective of analog circuit design. Firstly, it surpasses human-optimized values within the given parameter ranges, providing enhanced performance. Secondly, it offers substantial simplicity and reduced computational requirements compared to contemporary deep learning models, which makes it more effective as the number of cascaded circuit components increases. Lastly, it promotes light, versatile, and automated optimization for various analog circuit designs, aligning with research on model compression such as pruning and bit quantization due to the increasing complexity of deep learning models [18].

As circuit design complexity becomes more apparent, and the need for light, versatile, and efficient analog circuit optimization grows, the algorithm in this research stands as a valuable contribution to this field.

References

[1] Higuchi, T., Iwata, M., Keymeulen, D., Sakanashi, H., Murakawa, M., Kajitani, I., Takahashi, E., Toda, K., Salami, N., Kajihara, N., *et al.*: Real-world applications of analog and digital evolvable hardware. IEEE transactions on evolutionary computation **3**(3), 220–235 (1999)

[2] El-Turky, F., Perry, E.E.: Blades: An artificial intelligence approach to analog circuit design. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **8**(6), 680–692 (1989)

[3] Harjani, R., Rutenbar, R.A., Carley, L.R.: Oasys: A framework for analog circuit synthesis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **8**(12), 1247–1266 (1989)

[4] Lberni, A., Marktani, M.A., Ahaitouf, A., Ahaitouf, A.: Adaptation of the whale optimization algorithm to the optimal sizing of analog integrated circuit: Low voltage amplifier performances. In: 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), pp. 1–6 (2020). IEEE

[5] Xu, H., Ding, Y.: Optimizing method for analog circuit design using adaptive immune genetic algorithm. In: 2009 Fourth International Conference on Frontier of Computer Science and Technology, pp. 359–363 (2009). IEEE

[6] Biondi, T., Ciccazzo, A., Cutello, V., D'Antona, S., Nicosia, G., Spinella, S.: Multi-objective evolutionary algorithms and pattern search methods for circuit design problems. J. Univers. Comput. Sci. **12**(4), 432–449 (2006)

[7] Razavi, B.: RF Microelectronics, 2nd Ed. Prentice-Hall, Upper Saddle River, NJ, USA (2011)

[8] Bong, H., Cho, K., Seo, Y.: Automation framework for digital circuit design and verification. In: 2022 19th International SoC Design Conference (ISOCC), pp. 263–264 (2022). IEEE

[9] Song, I., Cho, M.-K., Oakley, M.A., Ildefonso, A., Ju, I., Buchner, S.P., McMorrow, D., Paki, P., Cressler, J.D.: On the application of inverse-mode sige hbts in rf receivers for the mitigation of single-event transients. IEEE transactions on Nuclear Science **64**(5), 1142–1150 (2017)

[10] Albataineh, Z., Moheidat, J., Hamada, Y.: Design of high gain 2.4 ghz cmos lna amplifier for wireless sensor network applications. In: 2017 10th Jordanian International Electrical and Electronics Engineering Conference (JIEEEC), pp. 1–5 (2017). IEEE

[11] Asharani, M., Nagabhushan, M.: A 2.4 ghz algorithmic design of cmos lna in

180nm technology. In: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 2579–2582 (2018). IEEE

[12] Yu, Y.-H., Chen, Y.-J.E., Heo, D.: A 0.6-v low power uwb cmos lna. IEEE microwave and wireless components letters **17**(3), 229–231 (2007)

[13] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin, Heidelberg (1996)

[14] Bao, Z., Watanabe, T.: A new approach for circuit design optimization using genetic algorithm. In: 2008 International SoC Design Conference, vol. 1, p. 383 (2008). IEEE

[15] Jaradat, Y., Masoud, M., Jannoud, I., Manasrah, A., Zerek, A.: Comparison of genetic algorithm crossover operators on wsn lifetime. In: 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), pp. 356–360 (2022). IEEE

[16] Feng, S., Wang, X.: Research on fault diagnosis of mixed-signal circuits based on genetic algorithms. In: 2012 International Conference on Computer Science and Electronics Engineering, vol. 3, pp. 12–15 (2012). IEEE

[17] Karabayir, I., Akbilgic, O., Tas, N.: A novel learning algorithm to optimize deep neural networks: evolved gradient direction optimizer (evgo). IEEE transactions on neural networks and learning systems **32**(2), 685–694 (2020)

[18] Deng, L., Li, G., Han, S., Shi, L., Xie, Y.: Model compression and hardware acceleration for neural networks: A comprehensive survey. Proceedings of the IEEE **108**(4), 485–532 (2020)