A Survey on Smart Optimisation Techniques for 6G-oriented Integrated Circuits Design

Thang Quoc Nguyen, Trang Hoang, Lihong Zhang, *Senior, IEEE*, Octavia A. Dobre, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

Abstract

With the rapid development of next-generation wireless communications, there is a growing demand for high-quality integrated circuits (ICs), particularly analog ICs, which play a pivotal role for the full roll-out sixth-generation (6G) technology. So far, the IC design has been performed through manual approaches which sometimes results in time-consuming turnaround, especially the sizing phase of analog IC design. In order to make the IC design process much faster, recently automated methods for optimizing IC design has gained a lot of attention. From this perspective, this paper aims at providing a survey of the most recent works on optimization strategies for analog IC sizing, as well as a related categorization into two main categories: analytical methods and simulation-based methods. A further sub-classification within the realm of simulation-based methods is also provided by dividing the core mathematical principles into three major sub-methods: Bayesian-based, metaheuristic-based, and reinforcement-learning-based techniques. In addition, with the main aim of providing insights on the utilization of optimization algorithms for the IC sizing process, we present a case study involving the utilization of various metaheuristic algorithms in the design of a bandgap reference circuit - an essential analog IC component. The paper is concluded by highlighting potential future research directions in the field of analog IC design optimization and automation, which include exploration of multi-

T. Q. Nguyen, L. Zhang, O. Dobre and T. Q. Duong are with the Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada (e-mail: {nqthang, lzhang, odobre, tduong}@mun.ca

T. Hoang is with Department of Electronics, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT) and also with Vietnam National University Ho Chi Minh City, Vietnam (e-mail: hoangtrang@hcmut.edu.vn).

This work was supported by Vietnam National University Ho Chi Minh City (VNU-HCM) under Grant DS2023-20-03. Corresponding author is Trang Hoang.

agent reinforcement learning, integration of quantum computing, and further development of full-flow automated tools for analog IC design.

Index Terms

Analog IC optimization, Simulation-based analog IC optimization, Bayesian optimization, Meta-heuristic optimization, Reinforcement learning, Quantum computing.

I. Introduction

Nowadays, we are witnessing how wireless communication, thanks to the development of high-speed wireless links, such as Bluetooth low energy (BLE), Wi-Fi, ZigBee, and LoRa, has reached a level of ubiquity. This is comparable to the state of electric power distribution, which facilitates seamless connections among various electronics devices, including laptops, cameras, phones, and any type of domestic appliances [1]. This surge in wireless communication can be attributed to several converging factors. Primarily, the continuous advancement in the quantity and quality of electronic circuits and components has played a pivotal role in this development process. In fact, recent advances in Complementary Metal-Oxide-Semiconductor (CMOS) technology have significantly enhanced the integrity of CMOS integrated circuits (ICs), particularly analog ICs, which are responsible for processing and transmitting analog signals, especially those in the radio frequency (RF) spectrum. However, maintaining such integrity enhancement along with the growing demand for high-quality integrated circuits to meet modern communication requirements, such as ultra-high accuracy, ultra-low latency, and ultra-high speed requirements in sixth-generation (6G) network communication [2], is making the design of analog circuits increasingly intricate and complex.

In general, the process of designing analog ICs follows a structured sequence comprising three main phases: topology design, transistor sizing, and physical design. The first phase involves converting the specified design requirements into a circuit structure. This strongly depends on the designer expertise in choosing and connecting suitable components such as MOSFETs, resistors, and capacitors, in order to establish the basic framework of the circuit. Subsequent to the topology design there is the crucial phase of transistor sizing, where the designer meticulously adjusts the dimensions (width and length) of each transistor in the selected architecture. It is worth mentioning that although initial estimates involve mathematical analysis, complex and non-linear relationships among component parameters and performance metrics often require an iterative method. This is due to the limitations of existing mathematical models in fully capturing the intricate interactions between these elements. Therefore, designers rely on a trial-and-error approach, beginning with calculated size approximations and then refining them through extensive simulations. This iterative procedure includes the modification of transistor sizes on the basis of the designer's experience and expertise to ensure that the overall circuit satisfies the specified performance requirement. Finally the physical design phase involves the transformation of the optimized circuit diagram into a physical layout suitable for silicon chip fabrication. This layout complies with predefined design regulations and guarantees manufacturability.

Between the three aforementioned phases in the analog IC design process, the transistor sizing phase presents a significant bottleneck in the analog IC design flow. Indeed, due to the complex interplay between component characteristics and circuit performance, coupled with the limitations of purely mathematical modeling, this phase can result to be time-consuming and require significant expertise. This extended design cycle would potentially negatively impact on the IC production businesses since it can cause potential market share losses due to delayed product launches. Indeed, as highlighted in [3], IC design businesses can lose up to 14% of market share if products are introduced four weeks late. In order to accelerate the analog IC design process and increase human effort in the sizing phase, numerous academic and commercial entities are actively researching alternative and faster methods to automate circuit design, which are mainly based on the usage of optimization algorithms. These algorithms offer substantial improvement over traditional methods by both reducing the time and effort required by designers and enhancing the overall quality of the final product. In fact, in contrast to digital circuit design that can be automated using Electronics Automation Design (EDA) tools, analog circuit design heavily depends on the designer expertise and experience, which makes it a time-consuming and labor-intensive process. This disparity between analog and digital design can be attributed to various factors. Firstly, analog circuits encompass a broader design space in terms of device size and topology compared to digital circuits, which predominantly rely on standard cell construction. Consequently, achieving desirable outcomes in analog design necessitates intricate and nuanced approaches. Secondly, the specifications governing analog design vary across different applications, posing challenges in devising a standardized framework for assessing and enhancing various analog designs. Last but not least, analog signals are inherently more vulnerable to noise and variations stemming from process, voltage, and temperature (PVT) fluctuations. Consequently, validating and verifying analog designs entail additional efforts to mitigate the impact of these factors [4].

Based on the discussion above, this paper aims at providing a survey about the recent research contributions on the topic of analog IC design optimization with a related categorization as summarized in Figure 1. More specifically, the main contributions of this article can be summarized as follows:

- Introduction of 6G performance requirements and how these requirements will imply the development of analog IC.
- Explanation of the problem formation process, in which the analog IC design specifications is translated into the language of an optimization problem.
- Classification of methods utilized to address analog IC optimization problems into two categories: analytical-based and simulation-based. Further sub-categorization of simulation-based methods into Bayesian-based, metaheuristic-based, and reinforcement-learning-based approaches.
- Illustration of a case study demonstrating the application of optimization algorithms in addressing an analog IC design challenge.
- Proposing potential research directions to address remaining challenges in the field of analog IC design optimization.

The rest of the paper is organized as follows. Section II provides a brief overview about 6G performance requirements and how these will impact on IC design and fabbrication. An overview of the optimization problem in IC design, detailing the formulation of an optimization problem in canonical form from IC design specifications is provided in Section III. In Section IV, we present a comprehensive summary of recent methodologies employed to address analog IC optimization problems, encompassing three leading approaches: Bayesian-based methods, metaheuristic-based methods, and reinforcement-learning-based methods. Section V presents a case study illustrating the application of five metaheuristic optimization algorithms as a demonstrative example of employing optimization algorithms in analog IC design. Potential research directions within the realm of analog IC design optimization, including the exploration of multi-agent reinforcement learning, quantum computing applications, and the prospective advancement of full-flow analog

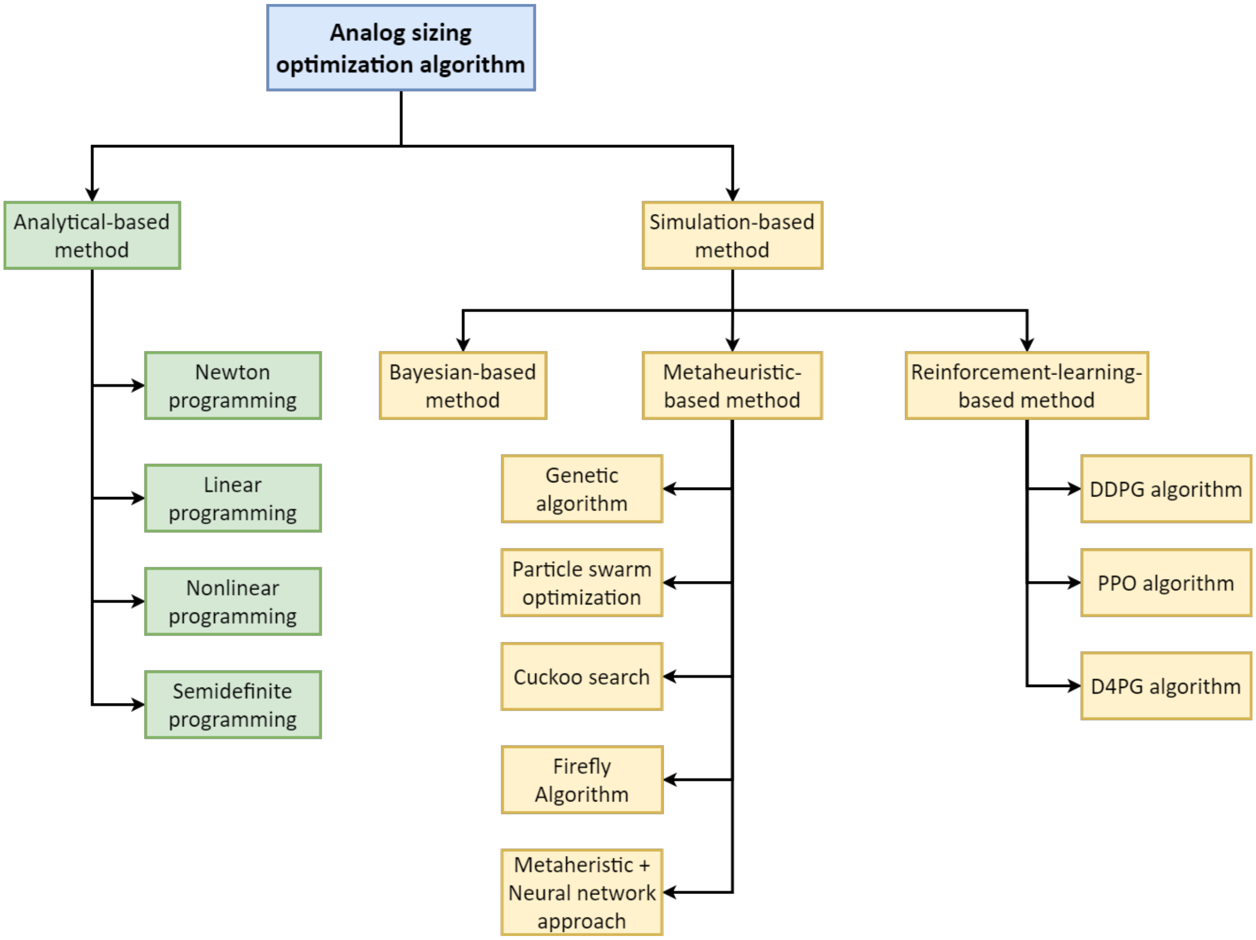

Fig. 1. Categorization of analog IC optimization techniques

IC design automation tools, are deliniated in Section VI. Finally, Section VII offers concluding remarks for the paper.

II. 6G Requirements Impact on IC Design

This section provides a brief overview about the main key performance indicators (KPI) expected to be delivered with the deployment of the 6G wireless networks. Subsequently, it highlights which are are the main challenges in the context of IC design that might hamper the full roll-out of 6G related technologies.

A. 6G wireless technology: Aims and requirements

Unprecedented development in wireless technologies has been observed over the last two decades. This has led to the constant deployment and diffusion of innovative services based on the concept of Internet-of-Things (IoT) communications. The IoT communication paradigm is mainly based on the interconnection of multiple wireless devices, such as smartphones, wearable electronic devices, autonomous vehicles, drones, and robots. These devices can communicate either in a peer-to-peer basis or with an edge/cloud service to provide a plethora of new use cases and services, including extended reality (ER), smart healthcare, intelligent transportation systems, smart industries, and global ubiquitous connectivity [5]. However, this widespread proliferation of wireless communication devices poses pressing challenges for mobile network operators. According to the International Telecommunication Union Radiocommunication Sector (ITU-R), it is expected that mobile data traffic will reach 5 zettabytes per month by 2030, a volume that cannot be accommodated by current 5G architectures [6]. Furthermore, in order to foster the deployment of IoT-oriented services, it will also be necessary to prioritize real-time communication with near-zero latency, where communication delays are less than 1 ms, and ultra-reliable transmission, i.e., ensure a communication error probability of less than 10^{-5}. This inevitably calls for the development of a wireless communication technology referred to as sixth-generation (6G) wireless networks. Indeed, compared to 5G architectures, 6G-based networks are expected to provide [7], [8]:

- 1 GHz operational bandwidth for operation in higher frequency bands like THz communications or optical wireless communications with a corresponding data rates up to 1 Gbps in downlink;
- Connection density up to 10^7 users per km^2;
- 10 μs of communication latency;
- Spectral efficiency up to 90 bps/Hz in downlink and 45 bps/Hz in uplink;
- Up to 1 Gbps/m^2 in some deployment scenarios such as indoor hot spots;
- A communication reliability in the order of 99.99999 %;

Then, one can easily notice how the achievement of these 6G-related KPI will ultimately address the upcoming issues in terms of increased networks capacity, improved reliability and reduced latency, which in turn will foster the deployment of new service aimed at facilitating our daily lives.

B. IC for 6G technology: Main challenges

As outlined in the discussion above, 6G technology is expected to bring about several improvements. However, achieving these improvements is dependent on the ICs necessary to build 6G-compliant devices. Indeed, compared to current devices oriented toward 4G/5G, it will be necessary to design new devices capable of operating at higher frequencies and possessing both higher computational capacity and energy efficiency [9]. This leads to a set of important challenges that need to be addressed to ensure the full deployment of 6G-oriented services and architectures. The most relevant and pressing challenges for the design of 6G-oriented devices are mainly related to the need for using communication frequencies in the THz bandwidth, extending up to visible light. These can be classified as outlined below

1) Operation High-Frequency: In order to meet 6G data rate requirements, it is expected that more frequency resources will be utilized. Consequently, due to spectrum scarcity, 6G communication devices are envisioned to operate at much higher frequencies than previous generations, potentially reaching into the terahertz (THz) range. However, this poses challenges such as high transmission loss, poor penetration, and limited non-line-of-sight (NLOS) coverage, which can significantly impact the quality of wireless signals. While beamforming and antenna steering techniques hold promise for improving signal quality in specific directions, designing highly efficient ICs for multi-antenna transceivers in the THz bandwidth is not straightforward. Precise dimensions, geometries, and special fabrication materials are necessary to realize antennas with efficient radiation and reception properties [10].

2) Signal Integrity and Interference Mitigation: In addition to utilizing higher frequencies, in order to increase the spectral efficiency 6G technologies require the adoption of modulation schemes such as Filter-Bank Multi-Carrier (FBMC), Universal-Filtered Multi-Carrier (UFMC), and Generalized Frequency Division Multiplexing (GFDM). These modulation schemes are more complex compared to those currently used in 4G/5G systems, such as Code Division Multiple Access (CDMA), Orthogonal Frequency Division Multiple Access (OFDMA), and Spatial Division Multiple Access (SDMA). As a result, maintaining signal integrity while mitigating interference becomes increasingly challenging. IC design must incorporate techniques to minimize signal distortion, noise, and interference to ensure reliable communication.

3) Integration of Multiple Technologies: 6G networks are expected to integrate various technologies, including massive MIMO, millimeter-wave communication, terahertz communication, and AI-based signal processing. Designing ICs that can seamlessly integrate these technologies while meeting size, power, and performance constraints is a complex challenge.

4) Energy Efficiency: Given the expectation of up to 1 Tbps of peak data rate in 6G networks, the necessity for faster signal processing becomes evident. Consequently, there is a need to further reduce clock cycles for signal processing. However, this poses a challenge as it requires more energy for computation, particularly in portable devices that are often energy-constrained. Therefore, the implementation of high-speed and low-power ICs for bit and packet processing becomes crucial.

5) Cost and Manufacturing Complexity: Developing ICs for 6G networks involves advanced manufacturing processes and materials, increasing manufacturing complexity and cost. Indeed 6G ICs often necessitate the use of advanced semiconductor process nodes, typically consisting in using smaller transistor sizes able to provide higher performance. However, these cause an increase of manufacturing complexity and cost due to the intricacies of fabrication at smaller scales. Furthermore, the choice of materials for 6G ICs is critical for achieving desired performance characteristics such as high-speed operation, low power consumption, and reliability. This involve the usage of novel material combinations that can quickly escalate manufacturing costs and complexity due to the need for specialized equipment and processes. This means that designing cost-effective ICs that can be manufactured at scale while meeting stringent performance and reliability requirements is a significant challenge.

In summary, the deployment of 6G technology brings forth numerous improvements, contingent upon the development ICs that adhere to 6G standards. Transitioning from 4G/5G to 6G devices necessitates designing new hardware capable of operating at higher frequencies while enhancing computational capacity and energy efficiency. This transition poses several key challenges that must be addressed to ensure the successful deployment of 6G-oriented services and architectures. Addressing these challenges requires collaborative efforts among semiconductor manufacturers, research institutions, and regulatory bodies to drive innovation and advancements in IC design and manufacturing for the successful realization of 6G technology.

III. Problem modeling

In analog IC design tasks, the main goal is to determine the dimensions (width and length) of each transistor within the circuit to meet the predefined specifications. These specifications may involve optimizing or minimizing specific performance metrics while ensuring that others exceed or remain below certain thresholds. For instance, consider the design of a bandgap reference (BGR) circuit, which is employed to provide an accurate reference voltage for an entire chip. In this context, various metrics are used to assess the performance of a BGR circuit, such as power dissipation, temperature coefficient, power supply rejection ratio, and how the BGR output voltage changes in response to variations in the power supply voltage, i.e., line sensitivity. In the context of sensor ICs, the primary concerns are low power consumption and minimal supply line sensitivity [11]. In contrast, for BGRs intended for quantum computing circuits, stability in a broad temperature range, particularly at cryogenic temperatures, takes precedence, making the temperature coefficient the most critical factor [12].

In most cases, the task of analog IC design can be formulated as an optimization problem as illustrated in (1):

$$\begin{aligned} \arg \underset{\mathbf{x}}{min} \quad & [F_1(\mathbf{x}), F_2(\mathbf{x}), ..., F_m(\mathbf{x})] && \text{(1a)} \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, i = 1, 2, .., n && \text{(1b)} \\ & g_j(\mathbf{x}) = 0, j = 1, 2, .., k && \text{(1c)} \\ & \mathbf{x} \in S && \text{(1d)} \end{aligned}$$

where:

- $\mathbf{x} \in \mathbb{R}$ represents the design variables, which can include the width and length of transistors in the circuit, as well as resistance and capacitance values.
- $F_1(\mathbf{x}), F_2(\mathbf{x}), ..., F_m(\mathbf{x})$ are the m functions that need to be optimized
- $f_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are unequal and equal constraints, representing the circuit specifications.
- $S \subset \mathbb{R}^+$ is the design space, limited by the upper bound and the lower bound of each design variable.

When $m = 1$, the optimization problem is single-objective, while if $m > 1$, the problem is called a multi-objective optimization problem. In contrast to single-objective optimization, multi-objective optimization problems present a more nuanced challenge. Indeed, obtaining a solution able to optimize all competing objective functions simultaneously is not always possible nor straightforward. This is mainly caused by the fact that these objectives are often in conflict. Then

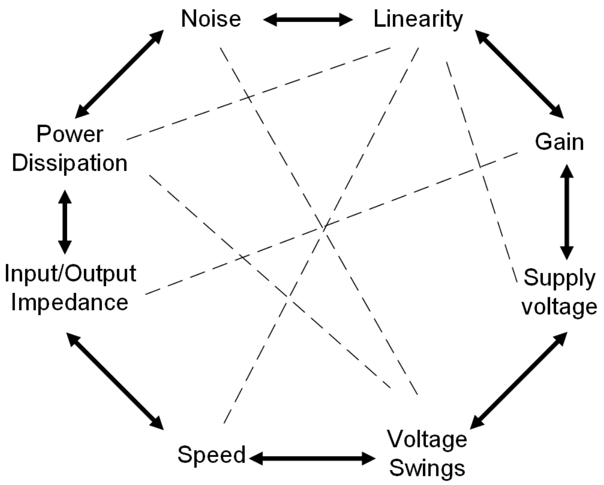

Fig. 2. Analog design octagon, reproduced from [14].

enhancing one may negatively affect the performance of others [13]. For instance, the design of a single-stage amplifier needs to take into account the interplay between different performance metrics of the single-stage amplifier as depicted in the ”analog design octagon” shown in Figure 2. In this illustration, the bold two-way arrows represent a conflict relationship, while the dashed lines represent a support relationship. For such a straightforward circuit, one can easily notice how the conflict and support dynamics among its eight performance metrics are intricate. This means that attempting to optimize more than one parameter, such as maximizing voltage swings and transition speed simultaneously, poses significant challenges. Indeed, in many cases, achieving the maximum value for both parameters concurrently is very difficult, if not impossible, to achieve. To reach a satisfactory solution for both objective functions, various techniques can be employed. Among these, the more common techniques include *i)* the weighted-sum technique which generates a new objective function by summing the original objective functions with weights, assigning a larger weighting coefficient to the more crucial objective function, *ii)* the usage of a utility function that generates a new objective function by multiplying or dividing the original objective functions, and *iii)* the ϵ-constrained technique which retains only one objective function and transforms the other objective functions into constraints.

IV. Methodology

In this section, we categorize the techniques suggested in prior works concerning the optimization of analog IC design into two primary groups: analytical-based methods and simulation-based methods. Additionally, within the simulation-based method category, we further classify into three main subgroups based on their mathematical principles: Bayesian-based methods, metaheuristic-based methods, and reinforcement-learning-based methods. Each of these methods is introduced along with key mathematical concepts and notable publications.

A. Analytical-based methods

In addressing the time-consuming and intricate process of analog IC sizing, a growing trend involves the utilization of optimization algorithms for automating circuit sizing. A particular type of approach is termed analytical-based method. This method utilizes a polynomial function to illustrate the connection between the geometrical size of the components and the circuit performance parameter. Subsequently, convex optimization techniques, such as Newton-based programming, linear and non-linear programming [15], [16], convex piecewise-linear fitting [17], linear matrix inequality relaxation [18], and semi-definite programming [19], are employed to solve the optimization problem. The essential advantage of the analytical-based method lies in its rapid execution. However, the effectiveness of this approach depends on the convex nature of the problem, requiring both the objective and constraint functions to exhibit convexity characteristics. Therefore, when formulating equations that link the physical size of components with the circuit performance parameters, a considerable amount of approximation might be required to transform the optimization problem into a convex form. This because as the size of each component starts to shrinks, the influence of secondary effects on the performance of the component itself, as well as on the overall circuit becomes more pronounced. As result, the approximate calculations start to become less reliable and unable to capture all the intricate details of high-order analog circuits [20], [21]. Therefore, the analytical-based method exhibits reduced effectiveness in the domain of analog circuit design [22].

B. Simulation-based methods

In order to address the limitations of analytical algorithms, several simulation-based methods have been investigated in recent decades. In general, the simulation-based method treats the circuit as a black box where having equations that directly describe the relationship between circuit performance metrics and components dimensions is not strictly necessary. In simulation-based method, the most essential body is the ”optimization core” - the machine learning algorithm

- which requires executing various SPICE [23] simulations to understand the relationship between geometrical sizes of components and circuit performance metrics, and discovers over the design space based on specific strategies (varying between algorithms) and simulation data. In this tutorial, we propose categorizing the previous work on the simulation-based method into 1) Bayesian-based method, 2) Metaheuristic-based method, and 3) Reinforcement-learning-based method. These methods are described below with some of the most relevant works presented in literature.

1) Bayesian-based methods: To enhance the effectiveness of optimization, the use of Bayesian optimization (BO) has been investigated. The BO algorithm, whose pseudo-code is shown in Algorithm 1, treats the circuit as a black box. Instead of using a detailed mathematical description, using sampled data the BO algorithm constructs a surrogate model which mimics the objective function and then provides predictions of its behavior at untried points. Based on the features provided by the surrogate model, the acquisition function decides the next sampling point by maximizing (or minimizing) itself.

Algorithm 1: Pseudo-code for BO algorithm

```
Initial Sampling
while not stopping do
    Find the position $\boldsymbol{x}$ at which the acquisition function is maximized
    Calculate $y = f(\boldsymbol{x})$
    Update the surrogate model
end
return best recorded $f(\boldsymbol{x})$
```

The most commonly utilized surrogate model is Gaussian process regression (GPR) [24]–[26]. Assuming a $d-$dimensional input design variable $\boldsymbol{x}$, we consider the unknown objective function as $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon = N(0, \sigma_n^2)$ represents the observation noise. We denote the sample dataset as $D = \{X, \boldsymbol{y}\}$, where $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}\}$ represents the set of design variables, and $\boldsymbol{y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, ..., \boldsymbol{y_N}\}$. By incorporating our prior beliefs regarding the performance of the unknown objective function using a predefined mean function $m(\boldsymbol{x})$ and kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the GPR model can offer a posterior distribution for any given location $\boldsymbol{x}^*$ as illustrated in Equation (2):

$$\begin{cases} \mu(\boldsymbol{x}^*) = m(\boldsymbol{x}^*) + k(\boldsymbol{x}^*, X) \times (K_N + \sigma_N^2 \times I)^{-1} \times (\boldsymbol{y} - \boldsymbol{m}), \\ \sigma^2(\boldsymbol{x}^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - k(\boldsymbol{x}^*, X) \times (K_N + \sigma_N^2 \times I)^{-1} \times k(X, \boldsymbol{x}^*). \end{cases} \tag{2}$$

where $\mu(\boldsymbol{x}^*)$ is the predictive mean, $\sigma^2(\boldsymbol{x}^*)$ is the uncertainty estimation, σ_N^2 denotes the variance of the Gaussian noise, $\boldsymbol{m} = (m(\boldsymbol{x}_1), m(\boldsymbol{x}_2), ..., m(\boldsymbol{x}_N))^T$ is the mean vector, $k(\boldsymbol{x}^*, X) = (k(\boldsymbol{x}^*, \boldsymbol{x}_1), k(\boldsymbol{x}^*, \boldsymbol{x}_2), ..., k(\boldsymbol{x}^*, \boldsymbol{x}_N))$, and K_N is the covariance matrix.

$$K_N = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

Within BO algorithms, the acquisition function plays a critical role in guiding the search process. This function essentially acts as a decision-maker, determining the most promising point for the next evaluation of the objective function $f(\boldsymbol{x})$. The acquisition function leverages the information provided by a surrogate model, which approximates the behavior of $f(\boldsymbol{x})$. Based on this surrogate model, the acquisition function identifies regions in the input space that warrant further exploration or exploitation. Areas where $f(\boldsymbol{x})$ has exhibited its best values or remain unexplored will receive high acquisition function values. On the contrary, regions where $f(\boldsymbol{x})$ has produced sub-optimal results or have already been sampled will receive low values from the acquisition function. By maximizing the acquisition function, the BO algorithm strategically selects the next sampling point with the highest potential to improve the objective function itself. This iterative process of sampling, updating the surrogate model, and maximizing the acquisition function allows BO to efficiently navigate the search space and converge towards the optimal solution for $f(\boldsymbol{x})$. Two of the most widely employed acquisition functions in BO are Probability of Improvement (PI) and Expected Improvement (EI):

- Probability of Improvement (PI): Calculated as $\mathrm{PI}(x) = \Phi(\lambda)$, where $\Phi(\lambda)$ represents the cumulative distribution function of the standard normal distribution. PI focuses on the likelihood of an arbitrary point $\boldsymbol{x}$ exceeding the current best objective function value based on the minimum observed value in the dataset.
- Expected Improvement (EI): Defined as $\mathrm{EI}(x) = \sigma(\boldsymbol{x})(\lambda\Phi(\lambda) + \Psi(\lambda))$, where $\Psi(\lambda)$ denotes the probability density function of the standard normal distribution. EI not only considers

the probability of improvement over the current best value but also factors in the magnitude of potential improvement.

Due to its reliance on the current best data point, the PI acquisition function tends to be conservative and biased toward exploitation, limiting its ability to identify the global optimum solution in scenarios with multiple local optima. Conversely, because the EI function incorporates the improvement magnitude into itself, EI search strategy is greedy, therefore it has slow convergence rate. To overcome this no-free-lunch characteristic of acquisition function, authors in [27] proposed a method called multi-objective acquisition function ensemble (MACE). This approach leverages the strengths of both PI and EI by sampling query points from the Pareto front formed by these acquisition functions, along with the lower confidence bound (LCB) function. The Pareto front represents a set of solutions where no objective can be improved without sacrificing another. By incorporating information from all three functions, MACE achieves a more balanced exploration-exploitation trade-off, leading to superior performance in analog IC optimization problems.

A notable limitation of the BO algorithm is that the existing acquisition functions like PI or EI are designed for unconstrained optimization problems, which are uncommon in the practical domain of analog IC design. In response to this challenge, a weighted-EI acquisition function, which integrates considerations of both the probability measure related to the objective function and the constraint functions has been proposed in [28], [29]. Moreover, authors in [27] introduced a two-stage algorithm to tackle constrained optimization problems. The first stage involves sampling data points from the Pareto front of the optimization problem described by Equation (3).

$$\arg \underset{\mathbf{x}}{min} \quad -\mathrm{PF}(\boldsymbol{x}), \sum_{i=1}^{N} \max\left(0, \mu_i(\boldsymbol{x})\right), \sum_{i=1}^{N} \max\left(0, \frac{\mu_i(\boldsymbol{x})}{\sigma_i(\boldsymbol{x})}\right) \tag{3}$$

where $\mathrm{PF}(\boldsymbol{x}) = \prod_{i=1}^{N} \Phi\left(-\frac{\mu_i(\boldsymbol{x})}{\sigma_i(\boldsymbol{x})}\right)$ is the probability of feasibility. This step prioritizes points that satisfy the constraints, effectively reducing sampling within invalid regions and increasing the likelihood of finding feasible solutions.

Subsequently, from these feasible points, the MACE algorithm is applied in conjunction with the objective function shown in Equation (3), as depicted in Equation (4).

$$\arg\underset{\mathbf{x}}{min} \quad \mathrm{LCB}(\boldsymbol{x}), -\mathrm{PI}(\boldsymbol{x}), \mathrm{EI}(\boldsymbol{x}), -\mathrm{PF}(\boldsymbol{x}), \sum_{i=1}^{N} \max\left(0, \mu_i(\boldsymbol{x})\right), \sum_{i=1}^{N} \max\left(0, \frac{\mu_i(\boldsymbol{x})}{\sigma_i(\boldsymbol{x})}\right) \tag{4}$$

To mitigate the likelihood of exploiting regions that do not meet constraints, only points that satisfy $\sum_{i=1}^{N} \max\left(0, \frac{\mu_i(\boldsymbol{x})}{\sigma_i(\boldsymbol{x})}\right) \leq 0.05$ are selected from the Pareto front of the optimization problem described in Equation (4).

Compared to analytical-based optimization methods, the Bayesian-based method offers several advantages in circuit optimization. It is more accurate than traditional analytical approaches. However, they also have limitations. As the design space grows, the computational cost of the algorithm becomes very high due to several factors such as the cubic training complexity and the square complexity of the GP model [30], as well as its inability to handle high-dimensional problems, known as the curse of dimensionality. To address the high-dimensional optimization challenges of the BO algorithm, a technique called circuit-BO (cBO) has been introduced in [26]. This cBO approach employs mutual analysis to identify the design variables that have the most impact on the target specifications, reducing the design space from D to $d < D$. The selected d variables are used to build the surrogate model, while the values of the remaining $D - d$ variables are determined using the g_m/I_D method. Another approach to reduce the dimensionality of the design space, proposed in [31], involves selecting the best candidate design region from multiple explored candidates. This method utilizes an enhanced GP to approximate the gradient and establish a 2-D subspace from the high-dimensional design space, followed by trust region-based derivative-free optimization (TR-DFO) for effective exploitation within the created subspace. Typically, addressing the curse of dimensionality in BO algorithms involves incorporating additional supplementary algorithms, which in turn increases the computational resources required.

2) Metaheuristic-based optimization methods: To address the limitations of analytical and Bayesian methods in the field of analog IC design, a contemporary optimization strategy, called the metaheuristic method, has been developed. This method combines SPICE simulation with metaheuristic algorithms, which are known as the most widely used optimization technique [32]. Several studies in this field have been published, most of them focused on combining SPICE with simulated annealing (SA) [33], genetic algorithm (GA) [20], [34]–[38], particle swarm optimization (PSO) [39]–[44], Cuckoo search (CS) [22], [45], and their hybrids [46]–[48].

In broad terms, the fundamental idea behind a metaheuristic algorithm involves employing a stochastic search across the design space in order to locate the best possible solution. The approach to performing this random search varies among different algorithms. The random search behavior of each metaheuristic algorithm, which is mimicked by the metaphor of group behavior of an animal in nature, or by the evolution process, generally consists of 3 steps:

Step 1: Generate multiple candidate coordinates for design variable vector. In the context of metaheuristic algorithm, each candidate coordinate of design variable vector is considered as a position of an individual, or a particle.

Step 2: Conduct an iterative procedure in which every particle is guided to explore the design space in order to find the global optimum solution. In this step, the algorithm consists of two main phases named simulation and evaluation. During the simulation phase, the coordinates of each particle, which represent the candidate values of design variables, are provided as input into the SPICE simulator to generate the necessary values for the objective and constraint functions. Following this, in the evaluation phase, the computed values of the objective and constraint functions are gathered by the algorithm to compute the fitness value for each particle. The algorithm then arranges the particles based on their fitness values, or stores the position where the best fitness value has been achieved thus far. Subsequently, the particles are relocated to new positions. The movement rules differ among specific algorithms, but generally, it can be characterized as a nonlinear mapping, as illustrated in Equation (5):

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{A}\left(\boldsymbol{x}_i^t,\ \ p(t),\ \epsilon(t),\ \boldsymbol{x_j}\right). \tag{5}$$

where $\boldsymbol{x}_i^t$ is the position of the i^{th} individual at the t^{th} iteration, $\boldsymbol{A}$ is the nonlinear mapping from $\boldsymbol{x}_i^t$ to $\boldsymbol{x}_i^{t+1}$, $p(t)$ is the vector of control parameters, $\epsilon(t)$ is the vector of random numbers, and $\boldsymbol{x}_j$ is the position of other factors that affect the movement of the particle, such as the current position of other particles, or the position at which the best fitness value has been found so far. Depending on specific algorithm, these terms are different, as illustrated in Table I. After moving particles to new position, the algorithm evaluates if the stopping criteria is met or not.

Step 3: The algorithm stops running and returns the global optimum solution has been found when the stopping criteria is met.

The process of using meta-heuristic algorithm to solve the optimization problem in analog IC

Algorithm	Control parameter	Random factor	Affected factors	Ref.
GA	Crossover probability p_c Mutation probability p_m	Random crossover position Random mutation position	Two randoly chosen individuals	[13], [20]
PSO	Inertia weight ω Acceleration coefficients c_1, c_2	Random numbers r_1, r_2	Current global best fitness position Best position found by each particle.	[13], [43], [44]
FA	Attractiveness β Light absorption γ	Gaussian distribution	Particles have higher fitness value	[13]
CS	Discovery probability p_a Step size α	Local walk: Random number ϵ Global walk: Lévy flight	Local walk: two randomly chosen particles Global walk: Current global best fitness position	[13]

TABLE I

COMPARISON OF METAHERISTIC ALGORITHMS

sizing is summarized in Algorithm 2.

Algorithm 2: Metaheuristic-based optimization method for analog IC sizing

```
Input  : Circuit netlist, variables boundaries, circuit specifications
Output: Global optimum solution
Initialize position for $N$ individuals /* Random or given */
while not stopping do
    /* Simulation phase */
    Call SPICE simulator to calculate the performance metrics values.
    /* Evaluation phase */
    Calculate fitness value for each individual based on performance metrics values.
    Ranking the individual based on their fitness value.
    Update positions based on Equation 5.
end
return individual has best fitness value.
```

A notable advantage of the this metaheuristic-based approach, compared to the analytical-based one, is its independence from a precise mathematical description [49]. Compared to the Bayesian-based method, the metaheuristic-based methods can be better in terms of dealing with high dimensional optimization problem, with lower computational cost compared to Bayesian-based method. However, as a result of the need to conduct multiple SPICE simulations, this metaheuristic-based approach requires more time and computational resources compared to the analytical-based method [20]. To address the time and computational demands of numerous SPICE simulations, some studies have proposed utilizing neural network models. By utilizing a dataset comprising pairs (ϵ, Γ), where ϵ represents the values of design variables and Γ denotes the values of circuit performance metrics, neural network algorithms are employed to construct a model that links device dimensions as inputs to performance metrics as outputs. Nevertheless, due to the nonlinear nature of the relationship between device dimensions and circuit performance metrics, the accuracy of the neural network model across the entire design space may be limited. Consequently, some studies suggested the use of the neural network model in specific regions of the design space. In this approach, an initial global search algorithm is utilized to identify regions that potentially contain optimal solutions, such as regions that satisfy all constraints [43]. During this stage, the SPICE simulator is employed to correlate the values of interest of design variables with corresponding performance metrics. Subsequently, the neural network algorithm gathers data around the local regions of interest and conducts training to construct a model for each region. Once the neural network models are established, a local search algorithm is employed to exploit these specific regions. During this phase, instead of utilizing the SPICE simulator, the trained neural network models are used to compute the performance metrics based on the design variables. Finally, the identified local optimal solutions are compared to determine the global optimum value. One key advantage of this approach is its efficiency in time. While it may take several hours to prepare adequate training data, once the model is trained, it only requires a few seconds to estimate the performance metrics from the design variables, making it significantly more time-efficient than conducting numerous independent SPICE simulations. Table II provides a summary of two papers that utilize the hybrid metaheuristic - neural network model approach.

Basically, metaheuristic algorithms are susceptible to becoming trapped in local optima. Increasing the number of individuals in a swarm, thus increasing swarm diversity, is a common strategy employed to enhance the likelihood of discovering a global optimum solution. However, in the domain of analog integrated circuit IC design, a larger population translates to a greater number of SPICE simulations, resulting in a rapid increase in execution time. To efficiently guide the algorithm towards the global optimum without compromising execution time, several research efforts have explored incorporating domain knowledge. This involves leveraging existing

	[37] **TCAD'20**	**[43]** **TCAS-I'23**
Circuit employed	Rail-to-rail op-amp Chebyshev band-pass filter	Op-amp
Algorithm for global search	GA	Grid search
Regions for local search	Around the best individual of each generation	In which all constraints are satisfied
Algorithm for local estimation	ANN	ANN
Algorithm for local search	Gradient-based	PSO
Speed enhancement compared to calling independent SPICE simulations	×4	×2.9 to ×75

TABLE II

SUMMARIZE TWO PAPERS ABOUT METAHEURISTIC - NEURAL NETWORK MODEL COMBINATION METHOD

knowledge about electronic circuits or the metaheuristic algorithm itself to steer individuals in the swarm towards the region containing the global optimum solution. Authors in [20] leveraged knowledge about the relationships between circuit components and performance metrics to identify elements (e.g. transistors) that impact a specific performance metric. This knowledge was then integrated into the GA mutation process. Specifically, if the unity gain bandwidth of a two-stage rail-to-rail operational amplifier falls below the desired value during an iteration, the value of the Miller capacitor decreases during mutation. In a similar way, an understanding about how reducing the inertia weight parameter in PSO can enhance the swarm's exploration ability, proposed a method called PSO - global exploration orienter (PSO-GEO), which involves exponentially decreasing this parameter across iterations has been presented in [44]. This adjustment aims to enhance the algorithm's ability to explore design spaces in the initial iterations to identify regions that may contain the global optimum, while subsequently exploiting the design space in later iterations to pinpoint precise solutions.

3) Reinforcement-learning-based methods: In order to address the limitations of conventional machine learning algorithms, reinforcement learning (RL) has showed great potential in the context of component sizing. Generally, a reinforcement learning algorithm consists of two

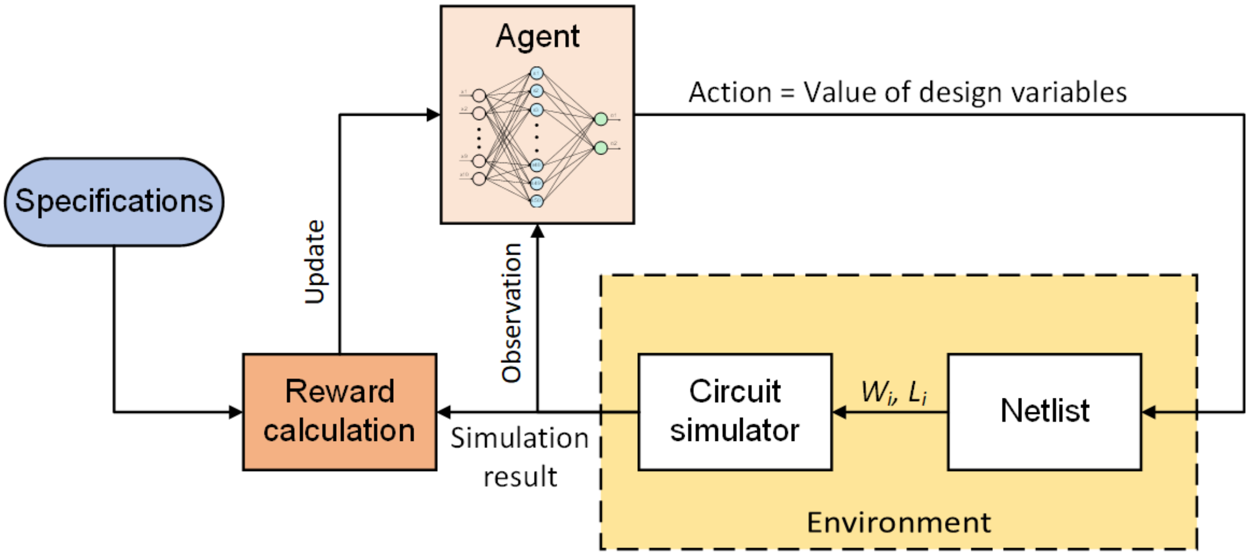

Fig. 3. Framework for utilizing reinforcement learning in optimization of analog IC sizing

elements: the agent and the environment. The agent, which is a machine learning algorithm, serves as a decision maker. Environment represents the problem or task that needs to be solved. The agent takes action on the environment, and the environment provides feedback to the agent in the form of reward. Based on its own policy and the reward received, the agent makes decisions about future interactions to maximize the cumulative reward.

The main framework of using RL in the analog IC design field is illustrated in Figure 3. In the realm of analog IC sizing, the circuit simulator serves as an environment. Based on the internal characteristics of the machine learning algorithm, agents generate the value of the design variables and then import these values into the netlist file. The circuit simulator runs simulation, resulting in performance metrics. Based on these performance metrics, the value of reward function is calculated.

RL algorithms can be categorized into two main approaches: model-based and model-free. The model-based method relies on an explicit environmental model that predicts action outcomes. This model is used to complement or replace direct interaction with the environment for policy learning. However, their reliance on the model limits their flexibility. On the contrary, algorithms that are model-free operate without any prior knowledge of the environment. This adaptability has resulted in its prevalence in recent applications of reinforcement learning. These techniques acquire knowledge through trial-and-error experiences, determining the best policy based on observed rewards. There are two primary categories of model-free algorithms: value-based and policy-based. Value-based algorithms establish the optimal policy by precisely calculating the value function for each state. Through interactions with the environment and sampling state-reward paths, the agent estimates the value function, which signifies the long-term anticipated reward for a specific state. Conversely, policy-based algorithms avoid modeling the value function and instead direct estimate the best policy. By parameterizing the policy with adjustable weights, these algorithms convert the learning process into a clear optimization challenge. Despite both methods sampling state-reward paths, policy-based algorithms explicitly enhance the policy by maximizing the average value function across all states. The primary limitation of policy-based RL algorithms is their elevated variance, which can result in training instabilities. On the other hand, value-based methods, although more consistent, encounter challenges in effectively representing continuous action spaces. By merging these divergent strategies, the actor-critic algorithm emerges as a potent solution. This technique needs the parameterization of both the policy (actor) and the value function (critic), allowing for optimal utilization of training data and ensuring stable convergence, surpassing policy-based approaches in efficiency and outperforming value-based methods in continuous and high-dimensional environments [50].

Multiple research studies have investigated the utilization of reinforcement learning in circuit sizing. One particular study, known as AutoCkt [51], [52], utilized a Proximal Policy Optimization (PPO) agent. The proposed method was implemented to assess a simple transimpedance amplifier and a two-stage operational amplifier in CMOS technology at a 45 nm node, and later on a different operational amplifier with a negative-g_m load in FinFET technology at 16 nm node. PPO is regarded for its stability and effectiveness, achieved by maintaining consistent policy updates throughout the training phase. Furthermore, its framework, which is illustrated in Figure 4, is suitable for situations with continuous or high-dimensional action spaces. Another significant study suggested a multi-step reinforcement learning method employing a deep deterministic policy gradient (DDPG) agent for designing two- and three-stage transimpedance amplifiers (TIA) [53]. This research captures both global (DC operating points, frequency response) and local (transistor characteristics) details at each step. With an actor-critic architecture, DDPG utilizes two networks. The actor network, which takes a state vector as input, produces an action vector. Meanwhile, the critic network, supplied with both state and action vectors, anticipates the reward value an agent can anticipate from its present and future actions. Unlike PPO, which

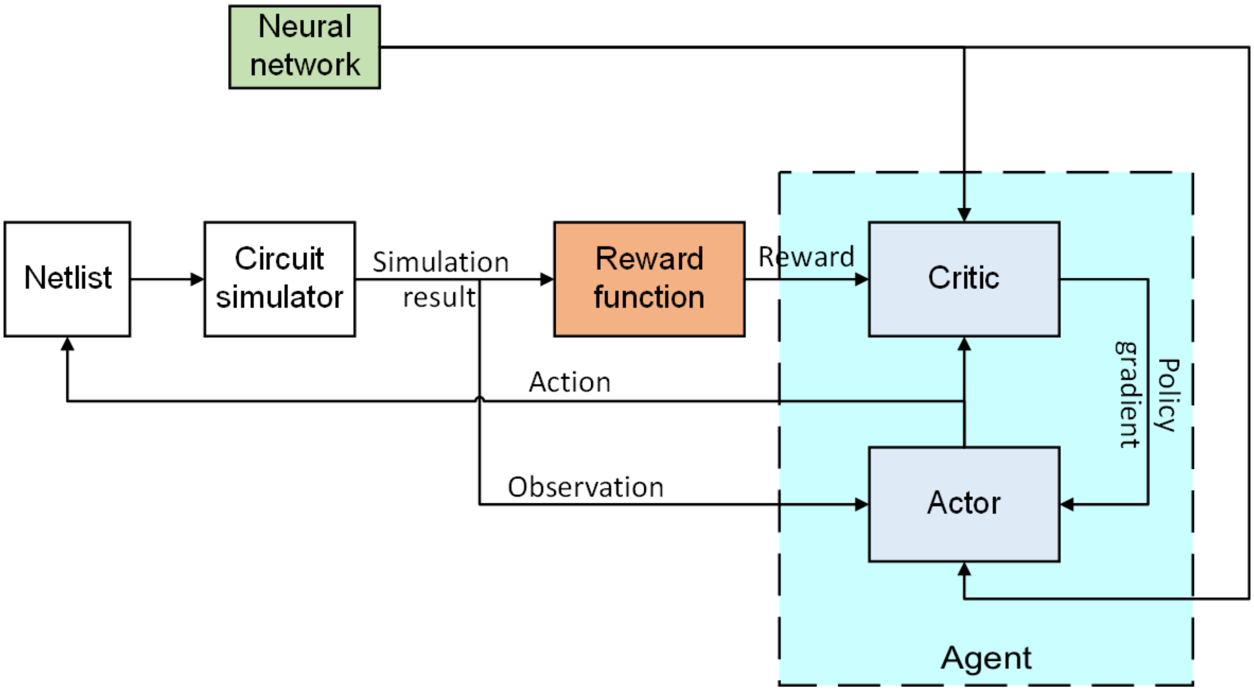

Fig. 4. Framework of DDPG agent employed in [53]–[55].

relies on parallel agent data collection during episodes to construct minibatch training sets, DDPG employs a single agent but utilizes a replay memory to recycle samples, resulting in a slower convergence rate but more efficiency in data sampling.

Following their aforementioned work, Wang et al. [54] continued to proposed a method using the graph convolutional neural network (GCN) as a function approximator in the DDPG agent, based on the nature that the circuit is also graph. In the proposed GCN, circuit components serve as nodes, and wires serve as edges. The l^{th} hidden layer of GCN is formulated as Equation 6:

$$H^l = \sigma(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2})H^{l-1}W^{l-1} \tag{6}$$

where $\widetilde{A} = A + I_N$ is summary of the adjacency matrix of the topology graph and the identity matrix, $\widetilde{D}_{ij} = \sum_j \widetilde{A}_{ij}$ is the diagonal degree matrix of $\widetilde{A}$, $\widetilde{W}^{l-1}$ is a layer-specific trainable weight matrix updated by DDPG agent.

Because the GCN agent described the knowledge related to the connection in the topology, it improves the learning ability of the RL model. The method also shows its ability to transfer the learned knowledge from the trained circuit to the circuit having the same topology but in different nodes. This method also shows the ability to transfer the trained knowledge from two-stage TIA to three-stage TIA and vice versa, resulting in reducing the number of training steps from 10000 to 300.

Some research articles based on the concept of DDPG-GCN have also been presented in literature. Authors in [55] applied the DDPG algorithm in conjunction with a relational graph convolutional neural network (RGCN) agent to address a multi-objective optimization challenge in the design of a low-dropout voltage regulator (LDO) circuit. A key advantage of RGCN over GCN is that each edge type in RGCN has its own independent weight matrix, allowing for a more precise representation of the circuit topology. This is particularly beneficial in capturing the distinct roles of different wires in an analog circuit, where some serve as DC biasing connections (akin to virtual ground) while others facilitate AC connections crucial for small-signal analysis. The optimization process using this approach requires 12 hours for the two-stage-op-amp LDO and 24 hours for the folded-cascode-op-amp LDO to obtain the optimal solution.

Another significant article suggesting to employ distributed distributional deep deterministic policy gradient (D4PG) as an agent has been presented in [56]. D4PG, a multi-agent variant of DDPG, enables parallel search capabilities. An advancement in D4PG, as opposed to DDPG, is its representation of the reward function in probability distribution form. This enhancement better models the uncertainty stemming from function approximation in a continuous environment. This, in turn, leads to improved training performance. Moreover, a novel aspect of this study, in contrast to the traditional D4PG method presented in [57], is the generation of multiple minibatches and the iterative updating of both the critic and actor networks. This strategy expedites the optimization process without adding extra time, as several updates can be performed concurrently with SPICE simulations. These enhancements have led to a decrease in the number of SPICE simulations necessary for D4PG to converge to the optimal solution, from 30,000 in the case of DDPG to 5,000.

In general, in the context of analog IC optimization, RL algorithms offer a compelling alternative to traditional metaheuristic approaches. Both explore the vast design space to identify optimal configurations. However, RL excels through its inherent memory capabilities. Unlike metaheuristic algorithms, RL agents possess a form of episodic memory, allowing them to retain information about past circuit states. This memory informs subsequent decisions, guiding the search process towards more promising regions of the design space. Furthermore, RL agents can leverage state-of-the-art neural networks as decision-making support tools. These neural networks

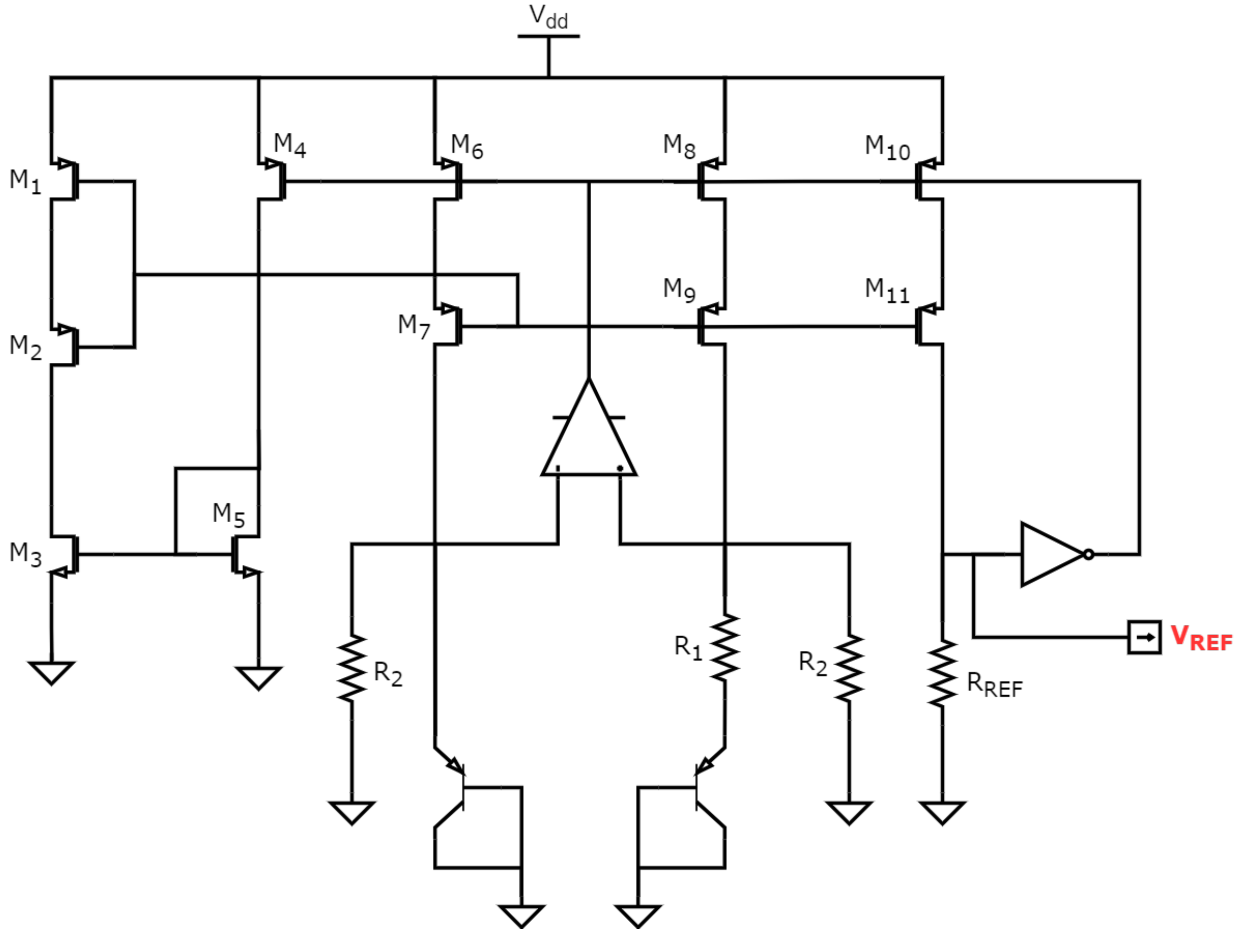

Fig. 5. Schematic of BGR circuit proposed by T. Hoang et al. [44]

can extract complex relationships from past circuit evaluations, enabling them to surpass the limitations of simpler metaheuristic approaches. This synergy between memory and advanced decision-making assistance positions RL as a powerful tool for optimizing analog IC design. Last but not least, the involvement of neural network in RL's agent also makes RL algorithm have the ability of transfer its learning to similar design problem, which saves time and computation cost.

V. Case study: Bandgap reference circuit design

In order to demonstrate the optimization process using a metaheuristic algorithm, we apply the metaheuristic optimization algorithm to enhance the power supply rejection ratio (PSRR) of the bandgap reference (BGR) circuit. The circuit topology for the BGR circuit, as proposed in [44], is depicted in Figure 5. The optimization framework utilized is depicted in Figure 6.

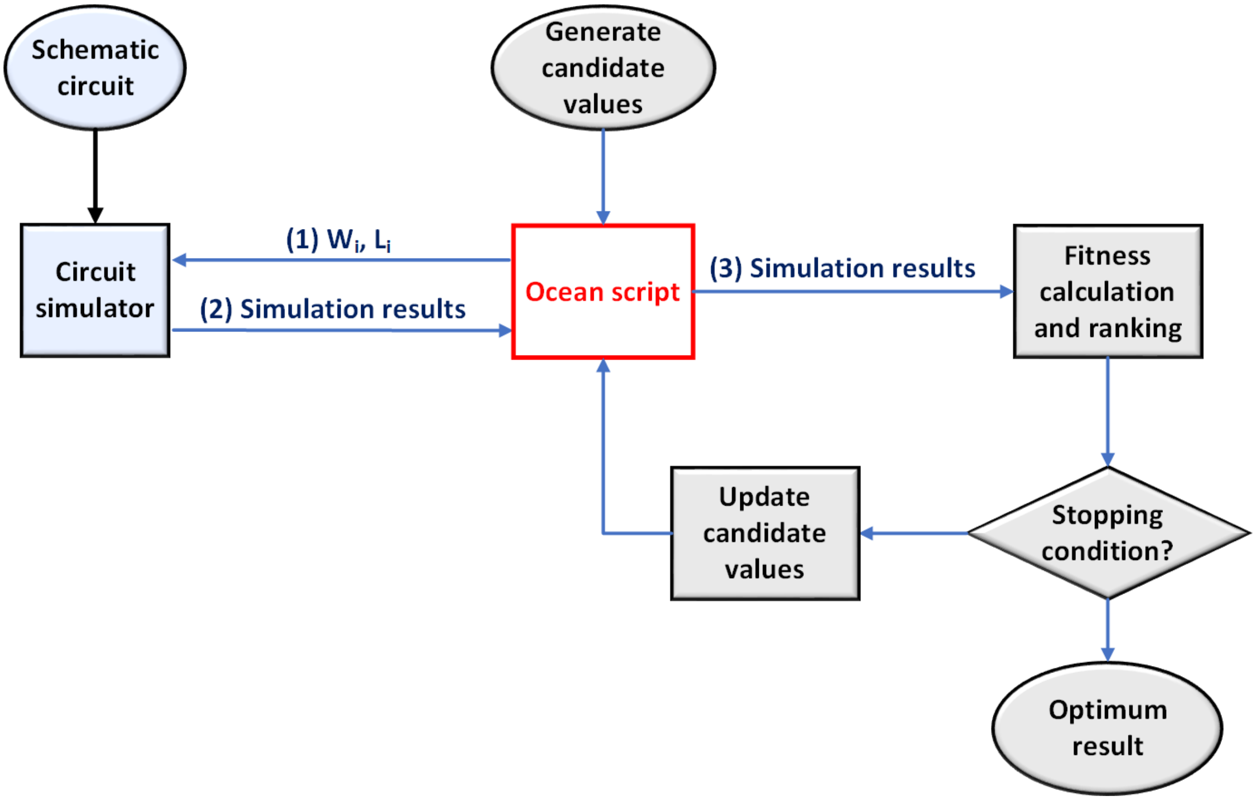

Fig. 6. Framework for implementing the metaheuristic algorithms in designing BGR circuit

This framework comprises two main components: the circuit block and the algorithm block. The circuit block includes the schematic representation of the BGR circuit, from which the netlist file is generated and subsequently imported into the Spectre simulator. In the algorithm block, the metaheuristic algorithms are implemented using the Python programming language. In this case study, five metaheuristic algorithms are utilized, which encompass the traditional PSO, the PSO-GEO introduced in [44], the gravitational particle swarm optimization algorithm (GPSOA) proposed in [58], the Cuckoo search (CS), and the firefly algorithm (FA). Facilitating the interaction between the circuit block and the algorithm block is a script written in the Ocean language. This script is employed to automate the execution of the Spectre simulator, facilitating the extraction of PSRR and constraint function values. To ensure a fair comparison among the five algorithms, identical initial positions and control parameters are established. For statistical robustness, each method undergoes 10 independent runs, each comprising 100 iterations.

The detailed optimization objective is shown in Table III. In this case study, we aim to maximize the value of PSRR, while keeping other metrics satisfied with their own constraints.

Metrics	Specification
PSRR	Maximize
$\mathrm{V_{REF}}$	$798\ \mathrm{V} \leq \mathrm{V_{REF}} \leq 802\ \mathrm{V}$
TC	$\leq 8\ \mathrm{ppm}/^{\circ}\mathrm{C}$
Loop gain @ DC	$\geq 40\ \mathrm{dB}$
Phase margin	$\geq 60^{\circ}$
Gain margin	$\geq 20\ \mathrm{dB}$
Power consumption	$\leq 400\ \mu\mathrm{W}$

TABLE III

SPECIFICATIONS FOR BGR CIRCUIT

To use the metaheuristic algorithm to solve this optimization problem, we formulate the fitness function depicted in Equation (7). This fitness function comprises two components: the negative value of the desired outcome (PSRR) and a penalty term. By minimizing the fitness function, the optimization algorithm effectively maximizes PSRR. The penalty term acts as a guardian, ensuring the compliance with predefined constraints throughout the search process. It starts at zero for each potential solution (particle) and is recalculated during evaluation. Each constraint is checked and, if violated, the penalty value increases by one. This progressive penalty discourages solutions that break the rules, guiding the search towards options that satisfy all constraints.

$$\mathrm{Fitness} = -\mathrm{PSRR} + 1000 \times \mathrm{penalty}. \tag{7}$$

Table IV summarizes the result collected from three algorithms after running 10 executions.

Table IV displays various optimization outcomes achieved by the five metaheuristic algorithms mentioned above. It is evident that, owing to the heuristic nature of these algorithms, the results vary not only between runs of the same algorithm but also across different algorithms. Each algorithm possesses its own strengths and weaknesses due to its unique mathematical characteristics. The three PSO-based algorithms exhibit lower success rates compared to the others, likely because the PSO algorithm tends to focus on exploiting the vicinity of the best fitness value (G^{*}) within the swarm. Conversely, the Cuckoo search algorithm and the firefly algorithm, influenced by Lévy flight and Gaussian distribution, respectively, tend to explore the

Execution	**Traditional PSO**	**GPSOA**	**PSO-GEO**	**CS**	**FA**
1	-97.592	901.214	-99.064	-86.582	-98.726
2	905.991	-97.595	-99.725	-97.762	-98.620
3	904.555	901.800	903.633	-97.565	-99.081
4	904.600	907.617	-100.234	-97.551	-98.528
5	898.464	903.237	899.591	-97.965	-98.612
6	-99.382	902.246	-99.794	-98.436	-97.865
7	899.296	900.008	-99.670	-98.523	-99.047
8	-100.299	905.598	-97.592	-93.487	-96.372
9	-97.700	-98.772	904.454	-98.752	-98.517
10	-98.000	-99.042	-99.941	-97.882	-98.372
Success rate	5/10	3/10	7/10	10/10	10/10
Minimum fitness value	-100.299	-99.042	-100.234	-98.752	-99.081
Mean value (success cases only)	-98.595	-98.470	-99.431	-96.451	-98.374
Std.dev (success cases only)	1.066	0.628	0.819	3.581	0.742

TABLE IV

Optimization result for 10 executions

design space rather than exploit it. This balance between exploration and exploitation is why the success rate of Cuckoo search surpasses that of PSO, and why traditional PSO and PSO-GEO can achieve solutions with superior fitness values and lower standard deviations compared to Cuckoo search, as shown in Table IV.Table IV displays various optimization outcomes achieved by the five metaheuristic algorithms mentioned above. It is evident that, owing to the heuristic nature of these algorithms, the results vary not only between runs of the same algorithm but also across different algorithms. Each algorithm possesses its own strengths and weaknesses due to its unique mathematical characteristics. The three PSO-based algorithms exhibit lower success rates compared to the others, likely because the PSO algorithm tends to focus on exploiting the vicinity of the best fitness value (G^*) within the swarm. Conversely, the Cuckoo search algorithm and the firefly algorithm, influenced by Lévy flight and Gaussian distribution, respectively, tend to explore the design space rather than exploit it. This balance between exploration and exploitation is why the success rate of Cuckoo search surpasses that of PSO, and why traditional PSO and PSO-GEO can achieve solutions with superior fitness values and lower standard deviations compared to Cuckoo search, as shown in Table IV.

VI. Potential for further research

It is promising to apply the aforementioned optimization techniques to assist the sizing problem when designing the analog IC. By applying these techniques, we can easily time- and labour-saving size all the components inside the circuit to satisfy predefined specifications without needing any expertise (the work of sizing which requires weeks to manually done now only need to wait about hours for executing the algorithm) and with high accuracy. Although remarkable progress has been made in this field, there are still some challenges to be overcome. The most notable challenge is the curse of dimensionality, which becomes more and more serious when applied to the radio frequency (RF) IC design. In the field of RF, we need to ensure that our circuit still works well over a wide frequency band and under multi-PVT corners. The specifications are made for not only in low frequency, but also in high frequency. In high frequency, the parasitic effect occurs, which encounters the effect of resistance, capacitance, and also inductance parasiting inside the body of CMOS components, as well as resistance, capacitance, and also inductance parasiting in the wires connecting the components. Furthermore, prior research has primarily addressed the design challenge of sizing intermediate-level circuits like multi-stage op-amps, BGR circuits, LDO circuits, or VCO circuits. Limited research has been conducted on sizing entire systems such as ADCs or phase-locked loops (PLLs) due to the challenge posed by the curse of dimensionality.

A. Multi-agent reinforcement learning

To address the curse of dimensionality challenge, various approaches can be explored as potential remedies. One such approach is multi-agent reinforcement learning, where multiple agents collaborate to maximize rewards. Multi-agent RL can be broadly categorized into centralized and decentralized types. Centralized RL involves all agents learning from a single policy and working together to optimize the total reward, while decentralized RL allows each agent to learn and act independently. An amalgamation of these two approaches is known as Centralized Training and Decentralized Execution (CTDE), which employs an actor-critic framework with a shared centralized critic among agents. Value-decomposition (VD) methods, a subset of CTDE algorithms, represent the collective Q function as a blend of individual agents' local Q functions, demonstrating remarkable performance in various multi-agent RL scenarios [59]. Despite the advancements in multi-agent RL research, the application of these techniques in analog IC design remains largely unexplored. There is a notable scarcity of studies investigating the utilization of RL in analog IC design and exploring the efficacy of different multi-agent strategies.

B. Quantum computing

Another potential solution to address the curse of dimensionality problem in analog IC sizing optimization is the application of an advanced computing technique - quantum computing. Quantum computing is well-known for its ability to solve complex high-computational optimization problems such as large numbers factorization and extensive searches [60] . Due to the inherent entanglement in quantum mechanics, a quantum circuit has the ability to operate on all n bits simultaneously: $U_f \sum |x\rangle |0\rangle \rightarrow \sum |x\rangle |f(x)\rangle$, which is called quantum parallelism [61], and also thanks to the superposition property, a quantum register can store 2^n states concurrently [62]. One of the well-known quantum search algorithm is Grover's search, which can effectively find the solution after $O\left(\sqrt{N}\right)$ operations compared to $O(N)$ operations of classical algorithms in average, has been used as a search engine in RL's agent to solve the optimization problem in wireless communication [63]–[65]. Grover's search RL can effectively deal with exploration and exploitation trade-off in high-dimensional problems. Another potential quantum technique for optimization is quantum annealing (QA), which offers a heuristic quantum optimization algorithm to find the ground state of Ising models. Compared to classical optimization algorithms such as genetic algorithms, particle swarm optimization, and differential evolution, the QA algorithm, which has been recently tested in solving the optimization problem in semiconductor manufacturing, can avoid being trapped in local minima and have the potential to find better global solutions due to the tunneling and superposition nature of qubits. Moreover, the QA algorithm is more robust to noise and other sources of errors than gate-based quantum algorithms [66].

C. Full-flow analog IC design optimization

Another potential direction for further research related to analog IC optimization is full-flow analog IC design automation, which may automate all three phases of analog IC design namely topology synthesis, sizing optimization and also layout optimization. Topology synthesis involves creating and enhancing the structure of an analog circuit according to specific design criteria. Transistor sizing phase is responsible for adjusting the dimensions of individual analog components to satisfy design specifications. Finally, the layout optimization aims to produce the layout of the analog circuit automatically. There are published articles that demonstrate the capacity of reinforcement learning in generating circuit topologies. Zhao and Zhang [67], [68] proposed using the Policy Gradient Neural Network (PGNN) for topology synthesis, and then use the NSGA-II algorithm for draft sizing. The method employed in these works required to prepare a building block library consisting 32 fundamental blocks such as common source, source follower, current mirror, and base on these fundamental block, more complicated blocks are built. In these works, the library construction requires time and costs, and in these works, the designed is such satisfied the predefined constraints, not optimization. Besides the building-block-based method, the graph-based method is introduced, such as Hong et el. [56] proposed using GA in synthesizing topology for level shifter circuit, and Lu et al. [29] proposed a method called bi-level BO, in which the BO algorithm is employed in both topology synthesis phase and sizing optimization phase. In these works, the circuit structure is described in graph structure, in which circuit components are represented by nodes and wires connecting them are represented by edge, and the optimization algorithm is employed to construct an optimum structure. Overall, these topology synthesis works, which, at the moment, are still in their development stages with simple circuits like the op-amp and level shifter.

VII. Conclusion

This survey has explored the fundamental concepts of formulating optimization problems from analog IC requirements and provided a comprehensive overview of state-of-the-art methods for solving analog IC sizing optimization problem. A comparison between analytical-based and simulation-based methods reveals that while analytical-based methods offer faster execution speeds, they struggle with complex optimization problems in modern IC design due to their dependence on problem convexity and the limited ability of small-signal-based mathematical equations to capture circuit behavior adequately. Consequently, simulation-based methods, including Bayesian-based, metaheuristic-based, and reinforcement-learning-based approaches, have become prevalent alternatives to traditional analytical methods. Despite Bayesian-based methods offering more accurate solutions and the capability to tackle non-convex optimization problems, they face limitations in handling high-dimensional optimization tasks and suffer from the computational overhead of Bayesian optimization algorithms and associated supplementary algorithms. On the contrary, metaheuristic-based algorithms exhibit lower computational costs and demonstrate superior performance in addressing complex problems. However, due to their heuristic nature, metaheuristic algorithms may encounter the challenge of getting stuck in local optima, especially in situations with numerous local optima. Furthermore, the heuristic nature of metaheuristic algorithms makes them less adaptable to changes in technology nodes. As a remedy, RL-based methods have been developed, taking advantage of neural networks within their agent architecture to improve strategy formulation and enable transfer learning. However, existing single-agent RL struggles as design space dimensions increase, necessitating further exploration of advanced optimization techniques such as multi-agent RL or quantum computing applications to address the curse of dimensionality problem effectively.

Acknowledgement

This work was supported by Vietnam National University Ho Chi Minh City (VNU-HCM) under Grant DS2023-20-03.

References

[1] P. P. Mercier, B. H. Calhoun, P.-H. P. Wang, A. Dissanayake, L. Zhang, D. A. Hall, and S. M. Bowers, "Low-power rf wake-up receivers: Analysis, tradeoffs, and design," *IEEE Open Journal of the Solid-State Circuits Society*, vol. 2, pp. 144–164, Oct. 2022.

[2] T. Q. Duong, L. D. Nguyen, B. Narottama, J. A. Ansere, D. V. Huynh, and H. Shin, "Quantum-Inspired Real-Time Optimization for 6G Networks: Opportunities, Challenges, and the Road Ahead," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1347–1359, Aug 2022.

[3] F. Zhang, *High-Speed Serial Buses in Embedded Systems*. Singapore: Springer, 2020.

[4] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong, X. Ning, Y. Ma, H. Yang, B. Yu, H. Yang, and Y. Wang, "Machine Learning for Electronic Design Automation: A Survey," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 26, no. 5, Jun. 2021.

[5] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, Feb. 2021.

[6] ITU-R, "IMT Traffic Estimates for the Years 2020 to 2030," July 2015.

[7] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 8, pp. 57 063–57 074, 2020.

[8] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[9] J. W. Lambrechts, S. Sinha, K. Sengupta, A. Bimana, S. Kadam, S. Bhandari, J. D. Preez, Z. Shao, X. Huang, Z. Liu, E. A. Karahan, T. Blundo, M. Allam, S. Ghozzy, J. Zhou, W. Fang, and J. Valliarampath, "Intelligent integrated circuits and systems for 5G/6G telecommunications," *IEEE Access*, vol. 12, pp. 21 402–21 419, 2024.

[10] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6g and beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.

[11] B. Park, Y. Ji, and J.-Y. Sim, "A 490-pW SAR Temperature Sensor With a Leakage-Based Bandgap-Vth Reference," *IEEE Trans. Circuits Syst. I, Exp. Briefs*, vol. 67, no. 9, pp. 1549–1553, 2020.

[12] H. Homulle, F. Sebastiano, and E. Charbon, "Deep-Cryogenic Voltage References in 40-nm CMOS," *IEEE Solid-State Circuits Lett.*, vol. 1, no. 5, pp. 110–113, 2018.

[13] X.-S. Yang, *Optimization Techniques and Applications with Examples*, 1st ed. Hoboken, NJ, USA: Wiley, 2018.

[14] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 2nd ed. New York: McGraw-Hill Education, 2017.

[15] J. Momoh, R. Adapa, and M. El-Hawary, "A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–104, 1999.

[16] J. Momoh, M. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. ii. newton, linear programming and interior point methods," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 105–111, 1999.

[17] J. Kim, J. Lee, L. Vandenberghe, and C.-K. K. Yang, "Techniques for improving the accuracy of geometric-programming based analog circuit design optimization," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, San Jose, CA, USA, 2004, pp. 863–870.

[18] S.-H. Lui, H.-K. Kwan, and N. Wong, "Analog circuit design by nonconvex polynomial optimization: Two design examples," *Int. J. Circuit Theor. Appl.*, vol. 38, no. 1, pp. 25–43, 2010.

[19] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *Proc. ACM/IEEE Design Autom. Conf.*, San Francisco, CA, USA, 2014, pp. 1–6.

[20] R. Zhou, P. Poechmueller, and Y. Wang, “An analog circuit design and optimization system with rule-guided genetic algorithm,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5182–5192, 2022.

[21] J. Tao, Y. Su, D. Zhou, X. Zeng, and X. Li, “Graph-constrained sparse performance modeling for analog circuit optimization via sdp relaxation,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 8, pp. 1385–1398, 2019.

[22] A. Fortes, L. A. da Silva Jr, R. A. Domanski, and A. Girard, “Two-Stage OTA Sizing Optimization Using Bio-Inspired Algorithms,” *J. Integr. Circuits Syst.*, vol. 14, no. 3, pp. 1–10, 2019.

[23] A. Vladimirescu, *The SPICE book*, 1st ed. New York, NY, USA: John Wiley & Sons, Jan. 1994.

[24] A. Sanabria-Borbón, S. Soto-Aguilar, J. Estrada-López, D. Allaire, and E. Sánchez-Sinencio, “Gaussian-process-based surrogate for optimization-aided and process-variations-aware analog circuit design,” *Electronics*, vol. 9, no. 4, p. 685, 2020.

[25] R. A. de Lima Moreto, C. E. Thomaz, and S. P. Gimenez, “Gaussian fitness functions for optimizing analog cmos integrated circuits,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1620–1632, 2017.

[26] C. Chen, H. Wang, X. Song, F. Liang, K. Wu, and T. Tao, “High-Dimensional Bayesian Optimization for Analog Integrated Circuit Sizing Based on Dropout and gm/ID Methodology,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4808–4820, 2022.

[27] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, “An Efficient Batch-Constrained Bayesian Optimization Approach for Analog Circuit Synthesis via Multiobjective Acquisition Ensemble,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 1, pp. 1–14, Jan. 2022.

[28] J. Lu, Y. Li, F. Yang, L. Shang, and X. Zeng, “High-Level Topology Synthesis Method for $|Delta$-Σ Modulators via Bi-Level Bayesian Optimization,” *IEEE Trans. Circuits Syst. I, Exp. Briefs*, vol. 70, no. 12, pp. 4389–4393, Jul. 2023.

[29] J. Lu, L. Lei, J. Huang, F. Yang, L. Shang, and X. Zeng, “Automatic op-amp generation from specification to layout,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 12, pp. 4378–4390, Dec. 2023.

[30] M. Choi, Y. Choi, K. Lee, and S. Kang, “Reinforcement learning-based analog circuit optimizer using gm/id for sizing,” in *Proc. ACM/IEEE Design Autom. Conf.*, San Francisco, CA, USA, 2023, pp. 1–6.

[31] T. Gu, W. Li, A. Zhao, Z. Bi, X. Li, F. Yang, C. Yan, W. Hu, D. Zhou, T. Cui, X. Liu, Z. Zhang, and X. Zeng, “Bbgp-sdfo: Batch bayesian and gaussian process enhanced subspace derivative free optimization for high-dimensional analog circuit synthesis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 43, no. 2, pp. 417–430, 2024.

[32] X.-S. Yang, S. Deb, M. Loomes, and M. Karamanoglu, “A framework for self-tuning optimization algorithm,” *Neural Comput. Appl.*, vol. 23, p. 2051–2057, 2013.

[33] G. Gielen, H. Walscharts, and W. Sansen, “Analog circuit design optimization based on symbolic simulation and simulated annealing,” *IEEE J. Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, 1990.

[34] P. Das and B. Jajodia, “Design automation of two-stage operational amplifier using multi-objective genetic algorithm and spice framework,” in *Proc. Int. Conf. Inventive Comput. Tech.*, Nepal, 2022, pp. 166–170.

[35] M. Taherzadeh-Sani, R. Lotfi, H. Zare-Hoseini, and O. Shoaei, “Design optimization of analog integrated circuits using simulation-based genetic algorithm,” in *Proc. Int. Stmp. Signals, Circuits Syst.*, vol. 1, Iasi, Romania, 2003, pp. 73–76 vol.1.

[36] O. B. Kchaou, A. Sallem, P. Pereira, M. Fakhfakh, and M. H. Fino, “Multi-objective sensitivity-based optimization of analog circuits exploiting nsga-ii front ranking,” in *Proc. Int. Conf. Synth. Model. Anal. Simul. Methods Appl. Circuit Design*, 2015, pp. 1–4.

[37] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, “An artificial neural network assisted optimization system for analog design space exploration,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2640–2653, 2020.

[38] T. S. Delwar, A. Siddique, U. Aras, and J. Y. Ryu, “A Design of Adaptive Genetic Algorithm-Based Optimized Power Amplifier for 5G Applications,” *Circuits. Syst. Signal Process.*, vol. 43, p. 2–21, Jan. 202.

[39] R. Rashid and N. Nambath, “Area Optimisation of Two Stage Miller Compensated Op-Amp in 65 nm Using Hybrid PSO,” *IEEE Trans. Circuits Syst. I, Exp. Briefs*, vol. 69, no. 1, pp. 199–203, 2022.

[40] A. Raj, S. Majumder, and G. P. Mishra, “Design of a CMOS based ring VCO using particle swarm optimisation,” *Analog Integr. Circ. Signal Process.*, Dec. 2023.

[41] R. Rashid and N. Nambath, “Hybrid Particle Swarm Optimization Algorithm for Area Minimization in 65 nm Technology,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Daegu, (South) Korea, 2021, pp. 1–5.

[42] K. G. Shreeharsha, R. K. Siddharth, M. H. Vasantha, and Y. B. N. Kumar, “Partition Bound Random Number-Based Particle Swarm Optimization for Analog Circuit Sizing,” *IEEE Access*, vol. 11, pp. 123 577–123 588, Nov. 2023.

[43] M. Fayazi, M. T. Taba, E. Afshari, and R. Dreslinski, “AnGeL: Fully-Automated Analog Circuit Generator Using a Neural Network Assisted Semi-Supervised Learning Approach,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 11, pp. 4516–4529, 2023.

[44] T. Hoang, T. N. Quoc, L. Zhang, and T. Q. Duong, “Novel Methods for Improved Particle Swarm Optimization in Designing the Bandgap Reference Circuit,” *IEEE Access*, vol. 11, pp. 139 964–139 978, 2023.

[45] A. Fortes, L. A. da Silva, and A. Girardi, “Low Power Bulk-Driven OTA Design Optimization Using Cuckoo Search Algorithm,” in *Proc. Symp. Integr. Circuits Syst. Design*, Ben Goncalves, Brazil, 2018, pp. 1–7.

[46] C. Li, F. You, T. Yao, J. Wang, W. Shi, J. Peng, and S. He, “Simulated Annealing Particle Swarm Optimization for High-Efficiency Power Amplifier Design,” *IEEE Trans. Microw. Theory Tech.*, vol. 69, no. 5, pp. 2494–2505, 2021.

[47] D. Joshi, S. Dash, A. Malhotra, P. V. Sai, R. Das, D. Sharma, and G. Trivedi, “Optimization of 2.4 ghz cmos low noise amplifier using hybrid particle swarm optimization with lévy flight,” in *Proc. Int. Conf. VLSI Design and Proc Int. Conf. Embedded Syst.*, Hyderabad, India, 2017, pp. 181–186.

[48] M. Barari, H. R. Karimi, and F. Razaghian, “Analog Circuit Design Optimization Based on Evolutionary Algorithms,” *Math. Problems Eng.*, vol. 2014, 2014.

[49] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley, and J. Hellums, “Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 6, pp. 703–717, 2000.

[50] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, “Deep reinforcement learning: A survey,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, Sept. 2022.

[51] K. Settaluri, A. Haj-Ali, Q. Huang, K. Hakhamaneshi, and B. Nikolic, “Autockt: Deep reinforcement learning of analog circuit designs,” in *Proc. Design, Automat. Test Europe Conf. Exhibit.*, Grenoble, France, 2020, pp. 490–495.

[52] K. Settaluri, Z. Liu, R. Khurana, A. Mirhaj, R. Jain, and B. Nikolic, “Automated Design of Analog Circuits Using Reinforcement Learning,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 9, pp. 2794–2807, 2022.

[53] H. Wang, J. Yang, H.-S. Lee, and S. Han, “Learning to Design Circuits,” in *Proc. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2018.

[54] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, “GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning,” in *Proc. ACM/IEEE Design Autom. Conf.*, San Francisco, CA, USA, Jul. 2020, pp. 1–6.

[55] Z. Li and A. C. Carusone, “Design and Optimization of Low-Dropout Voltage Regulator Using Relational Graph Neural

Network and Reinforcement Learning in Open-Source SKY130 Process," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, San Francisco, CA, USA, Oct. 2023, pp. 01–09.

[56] J. Hong, S. Kim, and D. Jeon, "An automatic circuit design framework for level shifter circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5169–5181, 2022.

[57] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," 2018. [Online]. Available: https://arxiv.org/abs/1804.08617

[58] S. Jiang, C. Zhang, W. Wu, and S. Chen, "Combined Economic and Emission Dispatch Problem of Wind-Thermal Power System Using Gravitational Particle Swarm Optimization Algorithm," *Math. Prob. Eng.*, 2019.

[59] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games," in *Proc. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022.

[60] T. Q. Duong, J. A. Ansere, B. Narottama, V. Sharma, O. A. Dobre, and H. Shin, "Quantum-Inspired Machine Learning for 6G: Fundamentals, Security, Resource Allocations, Challenges, and Future Research Directions," *IEEE Open J. Veh. Technol.*, vol. 3, pp. 375–387, Aug. 2022.

[61] M. Nakahara and T. Ohmi, *Quantum computing : from linear algebra to physical realizations*. USA: CRC Press, 2008.

[62] C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing*. Switzerland: Springer Cham, 2014.

[63] J. A. Ansere, T. Q. Duong, S. R. Khosravirad, V. Sharma, A. Masaracchia, and O. A. Dobre, "Quantum Deep Reinforcement Learning for 6G Mobile Edge Computing-based IoT Systems," in *Proc. Int. Wireless Commun. Mobile Comput.*, Marrakesh, Morocco, Jul. 2023, pp. 406–411.

[64] J. A. Ansere, D. T. Tran, O. A. Dobre, H. Shin, G. K. Karagiannidis, and T. Q. Duong, "Energy-Efficient Optimization for Mobile Edge Computing With Quantum Machine Learning," *IEEE Wireless Comm. Lett.*, vol. 13, no. 3, pp. 661–665, Mar. 2024.

[65] J. A. Ansere, E. Gyamfi, V. Sharma, H. Shin, O. A. Dobre, and T. Q. Duong, "Quantum Deep Reinforcement Learning for Dynamic Resource Allocation in Mobile Edge Computing-based IoT Systems," *IEEE Trans. Wireless Commun.*, pp. 1–1, Nov. 2023.

[66] P.-H. Fang, Y.-S. Chen, J.-S. Wu, and P. Yu, "Inverse Reticle Optimization With Quantum Annealing and Hybrid Solvers," *IEEE Access*, vol. 12, pp. 33 069–33 078, 2024.

[67] Z. Zhao and L. Zhang, "Deep reinforcement learning for analog circuit structure synthesis," in *Proc. Design, Automat. Test Europe Conf. Exhibit.*, Antwerp, Belgium, 2022, pp. 1157–1160.

[68] ——, "Analog Integrated Circuit Topology Synthesis With Deep Reinforcement Learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5138–5151, 2022.