IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 December 2025, accepted 25 January 2026, date of publication 30 January 2026, date of current version 6 February 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3659527

RESEARCH ARTICLE

Analog Integrated Circuit Optimization With an Enhanced Adaptive Quantum Genetic Algorithm

P. SRIDHAR AND HARISH MALLIKARJUN KITTUR, (Senior Member, IEEE)
School of Electronics Engineering, Vellore Institute of Technology, Vellore 632014, India
Corresponding author: Harish Mallikarjun Kittur (kittur@vit.ac.in)

ABSTRACT The optimization of analog integrated circuit synthesis is inherently a multi-objective problem, requiring the global optimum to satisfy all critical performance constraints. Classical genetic algorithms, although widely applied in analog design, often suffer from slow convergence and premature stagnation in high-dimensional design spaces. To overcome these limitations, this study proposes an Enhanced Adaptive Quantum Genetic Algorithm (EA-QGA), which integrates quantum computation principles with classical evolutionary strategies. Implemented using true quantum gate operations on IBM Qiskit simulators, the EA-QGA leverages a compact population, rapid convergence behavior, and global search efficiency inherent to quantum computing. The framework is evaluated on three benchmark analog circuits using 45 nm CMOS technology: a differential amplifier, a two-stage OTA, and a folded cascode OTA, and its performance, measured in terms of figure-of-merit (FoM), is compared against the Quantum-Inspired Genetic Algorithm (QIGA), which simulates quantum behavior through mathematical modeling. Experimental results show that, relative to QIGA, the proposed EA-QGA achieves FoM improvements of 0.62%, 3.33%, and 14.10% across the three evaluated circuits, while converging 4.6×, 5.8×, and 5.6× faster in generations and reducing runtime by 2.8×, 2.8×, and 2.4×, respectively. These results demonstrate that the proposed EA QGA consistently achieves globally optimal solutions with substantially improved computational efficiency, making it a promising candidate for high-performance analog IC design.

INDEX TERMS Analog circuit optimization, circuit sizing, evolutionary algorithm, quantum computing, quantum genetic algorithm.

I. INTRODUCTION

Designing analog integrated circuits (ICs) remains a complex challenge in modern electronics. Unlike digital circuit design, which has benefited from extensive automation, analog IC development continues to rely on the expertise of skilled engineers. This manual approach creates a growing bottleneck as circuit complexity increases, driven by demands for higher performance and miniaturization. The intricate behaviour of analog components, coupled with the vast and high-dimensional design space, makes circuit optimization both computationally intensive and time-consuming. Additionally, as analog and digital functionalities become more integrated within mixed signal System-on-Chip (SoC) architectures, efficient co-design strategies are crucial–yet analog circuit development often remains the most labour-intensive stage [1].

A key challenge within this process is circuit sizing: determining optimal component dimensions to meet performance targets. This is particularly demanding because of the complex, multi-dimensional relationship between component sizes and performance metrics, which necessitates balancing multiple performance trade-offs [2]. Circuit sizing aims to meet design requirements while optimizing application-specific metrics such as area, DC gain, slew rate, and power efficiency. Because no clear structural relationships connect these metrics to component dimensions, manually identifying optimal solutions is challenging. Conventional design methods begin with initial estimations based on simplified models or empirical formulas, followed by iterative refinement. This protracted process requires substantial designer expertise, underscoring the need for

The associate editor coordinating the review of this manuscript and approving it for publication was Yuh-Shyan Hwang.

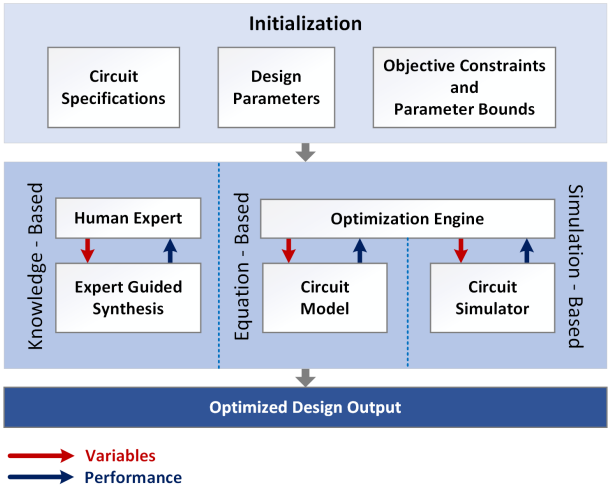


FIGURE 1. Overview of optimization categories.

automated sizing tools to enhance design efficiency, reduce effort, and accelerate analog IC innovation.

Manual analog circuit sizing's inherent complexities have spurred the development of automated methodologies, broadly categorized as knowledge-driven or optimization-driven, as illustrated in Fig. (1). Knowledge-driven methods, early automation attempts, capture expert insights via algorithms, rules, or mathematical formulations [3]. While valuable for initial exploration, they struggle with precisely encoding complex design knowledge, especially with advancing technologies. Simplified models and predefined equations can cause inaccuracies, requiring refinement. Furthermore, these methods are often circuit-specific and lack scalability. A newer look-up table approach [4] improves efficacy and precision but introduces speed-accuracy trade-offs, and the finite accuracy of look-up tables themselves can pose challenges for high-precision designs.

Building on these advancements, optimization-driven methods fall into two main categories: equation-based (model-driven) and simulation-based (black-box) optimization. These approaches iteratively refine design parameters, replacing fixed circuit plans. Their effectiveness depends on the optimization and evaluation mechanisms used to guide the process. Equation-based optimization expresses circuit performance and constraints as mathematical functions of design parameters, typically employing analytical expressions or symbolic models [5], [6]. While techniques such as geometric programming [7], [8] and g_m/I_D-based methodologies [9], [10] enhance model accuracy, capturing highly nonlinear effects and higher-order interactions in complex circuits remains a challenge. Simplifications in mathematical modeling, often required for tractability, can introduce inaccuracies, making these methods less reliable for circuits with intricate dependencies.

Consequently, simulation-based optimization offers an alternative, treating the circuit as a "black box" and using circuit simulators (e.g., SPICE) to evaluate performance for given design parameters. An optimization algorithm iteratively proposes parameter combinations, simulates the circuit for each, and refines its search based on the simulation results [11]. This approach is more versatile and accurate, relying on detailed simulations rather than simplified equations. Numerous techniques are employed, including evolutionary algorithms [11], [12], [13] such as Genetic Algorithm [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], Differential Evolution [26], Particle Swarm Optimization [27], [28], [29], [30], [31], and Firefly Algorithm [32], as well as Simulated Annealing [33], Seeker Optimization [34] and Bayesian Optimization [35]. Simulation-based methods are generally easier to implement than model-based approaches, requiring less complex mathematical derivations, and less specialized circuit expertise compared to knowledge-based approaches. Crucially, their use of detailed simulations provides greater accuracy and enables them to handle the complexities of modern analog circuits. A key strength lies in their ability to address complex circuit behavior without the simplifying assumptions inherent in model-based methods, allowing exploration of a wider design space and potentially more optimal solutions. However, their primary limitation is computational cost, as each circuit evaluation requires a simulation, which can be time-consuming, especially for large circuits or complex simulations.

A prominent evolutionary algorithm employed in analog circuit sizing is the Genetic Algorithm (GA) [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. GA leverages evolutionary principles to efficiently explore the design space, making it well-suited for this task. However, GA can be computationally demanding due to large population sizes and the numerous simulations that are often necessary. Furthermore, it can be prone to becoming trapped in local optima, converging to a satisfactory solution but not necessarily the globally optimal one. To mitigate these challenges, a new class of evolutionary algorithms has emerged: the Quantum Genetic Algorithm (QGA) [36], [37], blending quantum computing principles [38], [39] with the Genetic Algorithm (GA) framework. QGA represents a probabilistic approach to optimization, utilizing qubits to represent design parameters and enabling a more efficient exploration of the design space [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]. By leveraging the concurrent multiple states of qubits, QGA can explore a wider range of possibilities simultaneously, significantly improving upon classical GA. This enhanced exploration leads to faster convergence, a reduced risk of local optima, and ultimately, the discovery of superior circuit designs.

Despite these advantages, conventional QGA encounter significant challenges in analog circuit optimization. The design space is vast and consists of diverse parameter constraints, including transistor gate width, channel length, bias currents, voltages, and compensation components, all of which vary widely in range, making optimization more complex. Encoding such a large and heterogeneous solution

space into quantum populations is constrained by the limited number of qubits available in quantum simulators. While increasing qubit resources can alleviate this limitation, it incurs substantial computational costs, restricting practical scalability. As a result, quantum-inspired genetic algorithms (QIGAs) and quantum-based genetic algorithms (QBGAs) [50], [51], [52], which simulate quantum principles using classical computations, have gained prominence. Nevertheless, these algorithms fail to exploit true quantum advantages such as superposition and entanglement, limiting their effectiveness in solving high-dimensional optimization problems.

A further challenge in conventional QGAs is their reliance on lookup tables for determining rotation angles in genetic operations [41], [42], [43], [44], [45]. These fixed rotation angles result in inefficient searches and slower convergence, often leading to premature stagnation. While prior work has explored self-adaptive strategies, such as linearly decreasing rotation angles over generations [46] or sigmoid-based dynamic adjustments [51], these static schedules struggle to adapt to real-time changes in the fitness landscape, hindering search performance. Moreover, the fixed mutation probability used in conventional QGAs [47] across all generations increases the risk of premature convergence and limits exploration of the search space. In addition, existing quantum mutation strategies suffer from limited diversity due to their reliance on deterministic NOT gate swaps [46], [48], [49], as these operations merely exchange qubit states with certainty, failing to generate the stochastic fluctuations crucial for diverse exploration. Furthermore, the complexity and conditional, potentially erratic, fitness-dependent randomness inherent in discrete fitness-based mutation strategies lead to sensitivity to parameter tuning [51], and the instability of standard deviation-driven adaptive mutation rates can mislead convergence, particularly in small populations [52].

To overcome these challenges, obtaining the best optimal solution alone is not sufficient; instead, a more robust and adaptive methodology is required to achieve the true global optimum. A robust dimensionality reduction technique is essential to efficiently map large design spaces onto the available qubits. Additionally, a feedback-driven rotation angle adaptation mechanism is necessary to dynamically adjust gate operations based on the evolving optimization progress. Furthermore, an adaptive mutation strategy is crucial for maintaining population diversity and enhancing global search capabilities.

Therefore, this paper proposes a novel Enhanced Adaptive Quantum Genetic Algorithm (EA-QGA) that integrates the benefits of true quantum computing, genetic algorithms, and circuit simulation tools (SPICE) to optimize analog integrated circuit sizing and synthesis. The key contributions and distinguishing features of this work are outlined as follows:

- To address the challenge of mapping a large optimization space onto a limited number of qubits in quantum simulators, we propose a Principal Component Analysis (PCA)-based dimensionality reduction technique. This approach preserves the most meaningful optimization variables while enhancing the flexibility to explore the entire solution space more effectively. By reducing the dimensional complexity, the proposed method improves the accuracy of finding globally optimal solutions, mitigating constraints imposed by qubit limitations.
- Integrating QGA with SPICE allows for fitness evaluation of each quantum population based on multiple performance metrics, thus preserving the multi-objective nature of the solutions.
- We propose a novel Gradient-Based Quantum Gate Rotation (GBR) strategy that dynamically adjusts rotation angles using gradient-like feedback from fitness values, enabling adaptive search based on performance trends across generations.
- To enhance search space exploration and mitigate premature convergence, this approach employs an adaptive mutation strategy that dynamically adjusts mutation probability and strength based on population fitness diversity, specifically computed using the Coefficient of Variation (CV).

The structure of this paper unfolds as follows: Section II lays the theoretical groundwork, covering key aspects of analog circuit parameter optimization and quantum computing fundamentals. Section III details the proposed Enhanced Adaptive Quantum Genetic Algorithm, detailing its framework, population generation using PCA, fitness evaluation, and adaptive genetic operations, including quantum operations such as gradient-based gate rotation, crossover, and mutation. Section IV evaluates the performance of the proposed method on benchmark circuits, including a CMOS differential amplifier, a two-stage operational amplifier and a folded cascode amplifier. Finally, Section V concludes the paper with a summary of findings and future research directions.

II. THEORETICAL FRAMEWORK

A. ANALOG CIRCUIT PARAMETER OPTIMIZATION

The optimization of analog circuit parameters is a constrained optimization problem aimed at determining the optimal design variables to maximize a Figure of Merit (FoM) while adhering to performance constraints. The FoM (MHz*pF/mW), which quantifies power-bandwidth efficiency and is expressed as (1) according to [11], [20], and [21].

$$\text{FoM} = \frac{GBW * C_L}{Power} \tag{1}$$

where GBW is the gain bandwidth product (MHz), C_L is the load capacitance (pF), and Power is the total power dissipation (mW).

The design variables, represented as X, are constrained within upper bound (UB_i) and lower bound (LB_i) limits, forming a multi-dimensional search space S. The optimization

problem is mathematically formulated as (2)

$$\begin{aligned} &\max_{X \in S} \ \text{FoM}(X) \\ &\text{subject to} \quad g_i(X) \geq 0 \\ &\qquad\qquad h_j(X) = 0 \\ &\qquad\qquad LB_i \leq X_i \leq UB_i, \quad \forall i \in \{1, 2, \ldots, D\} \end{aligned} \tag{2}$$

where $g_i(X)$ represents inequality constraints ensuring required performance metrics, $h_i(X)$ denotes equality constraints, D is the total number of design parameters.

These constraints ensure that the optimized circuit meets specifications for gain, unity gain bandwidth, and slew rate while minimizing power consumption and area. To consistently enforce these constraints during optimization, a feasibility-based objective function is adopted, as defined in (3). This formulation assigns a valid FoM value only to candidates satisfying all constraints in (2), while infeasible solutions receive a penalized value, thereby guiding the search toward valid and high-performance analog circuit designs:

$$F(X) = \begin{cases} \text{FoM}(X), & \text{if } g_i(X) \geq 0 \ \forall i \text{ and } h_j(X) = 0 \ \forall j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

B. QUANTUM COMPUTING PRINCIPLES

Quantum computing is a transformative field that utilizes the principles of quantum mechanics to process information differently from classical computing.

1) QUANTUM STATE

In quantum mechanics, the state of a quantum system can be represented using vector notation or Dirac's bra-ket formalism. A ket (e.g., $|x\rangle$) denotes a column vector, while a bra (e.g., $\langle x|$) represents its conjugate transpose. The fundamental states of a quantum system are denoted as $|0\rangle$ and $|1\rangle$, corresponding to the standard basis vectors $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$, respectively. Any general quantum state exists as a linear combination (superposition) of these basis states and can be mathematically expressed as (4)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{4}$$

where α and β are complex numbers known as probability amplitudes, satisfying the normalization constraint (5). This normalization ensures that the probabilities of all possible measurement outcomes sum to one.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{5}$$

2) QUBIT

A qubit (quantum bit) serves as the basic unit of quantum information. Unlike a classical bit, which can only assume discrete values (0 or 1), a qubit can exist in a superposition of both states. When multiple qubits are combined, the overall quantum state is described by the tensor product of individual qubits, forming an exponentially larger state space. For an N-qubit system, the quantum state is represented as (6)

$$|\psi\rangle = \sum_{i=0}^{2^{N-1}} c_i |i\rangle \tag{6}$$

where c_i are complex probability amplitudes, and the summation spans all possible 2^N basis states.

Another key concept is entanglement, where multiple qubits become interdependent, allowing for correlated states that enhance computational capabilities. Quantum operations are performed using quantum gates, which manipulate qubits through unitary transformations, enabling the creation of superpositions and entangled states.

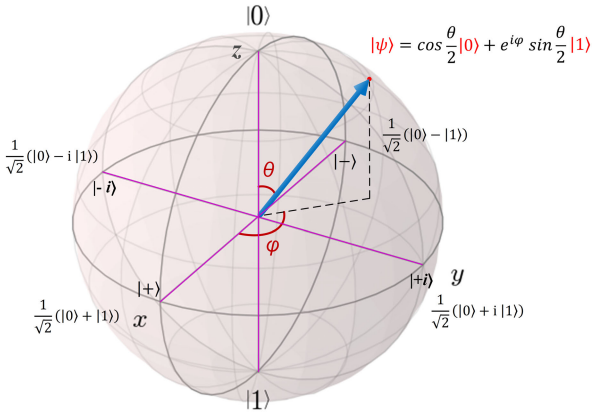


FIGURE 2. Visualization of qubit superposition and phase using the Bloch Sphere framework.

3) QUANTUM MEASUREMENT AND STATE COLLAPSE

Measurement in quantum mechanics extracts classical information from a qubit, causing its state to collapse to one of the measured basis states. With $|\psi\rangle$ as defined in equation (4), measuring this qubit in the computational basis yields the state $|0\rangle$ with probability $|\alpha|^2$ and the state $|1\rangle$ with probability $|\beta|^2$. Once measured, the superposition is lost, and the qubit collapses irreversibly into the observed state.

4) BLOCH SPHERE

A single qubit state can also be visualized on the Bloch sphere as shown in Fig. 2, where any quantum state is expressed in spherical coordinates $(r,\ \theta,\ \phi)$ as (7)

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{7}$$

where $\theta \in [0, \pi]$ represents the latitude, $\phi \in [0, 2\pi]$ represents the longitude.

This representation highlights the geometric nature of quantum states and how rotations on the sphere correspond to quantum operations.

5) QUANTUM GATES

Quantum gates are the operational core of quantum computing, facilitating transformations on qubits through unitary operations that inherently preserve quantum state probabilities. Unlike their classical counterparts, these gates are reversible, ensuring no computational information is lost. Fundamentally, they are represented by unitary matrices, satisfying the condition $UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U, and I is the identity matrix. These gates can be broadly classified into single-qubit and multi-qubit gates, each playing a distinct role in quantum computation.

Single − QubitGates : These gates act on individual qubits, modifying their states by flipping, introducing superposition, or applying phase shifts.

Pauli Gates (X, Y, Z): These matrices represent fundamental quantum transformations along the Bloch sphere's three axes. The Pauli-X gate, (X), acts as the quantum analogue of the classical NOT operation, swapping the basis states $|0\rangle$ and $|1\rangle$. The Pauli-Y gate, (Y), combines a bit-flip with a phase shift, effectively rotating the qubit around the Y-axis. The Pauli-Z gate, (Z), leaves $|0\rangle$ unchanged but applies a π phase shift to $|1\rangle$, impacting quantum interference. The matrix representations of the Pauli gates are:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{8}$$

Hadamard Gate (H): A fundamental operation in quantum computing, the Hadamard gate enables the creation of superposition. When applied to $|0\rangle$, it produces an equal superposition of $|0\rangle$ and $|1\rangle$:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{9}$$

This transformation ensures that measurement outcomes are probabilistically distributed, a crucial property for quantum parallelism.

Rotation Gates (R_x, R_y, R_z): These gates allow controlled rotations of a qubit around the Bloch sphere's principal axes, facilitating precise control of probability amplitudes. The mathematical representations of these rotation gates are as follows:

$$\begin{aligned} R_x(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\ R_y(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\ R_z(\theta) &= \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \end{aligned} \tag{10}$$

These rotational gates enable precise state control, which is crucial for quantum algorithms requiring variable phase adjustments

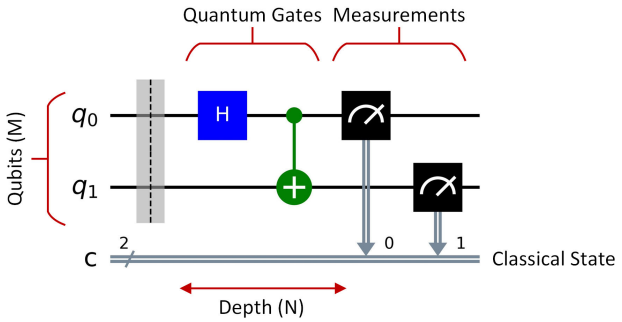


FIGURE 3. Quantum Circuit: This diagram represents a quantum circuit with qubits (horizontal lines) and circuit depth (horizontal progression). It includes a single-qubit Hadamard gate (H), a two-qubit CNOT gate, and measurement operations (double lines) that produce classical bit outputs–key components of quantum computation.

Multi−QubitGates : Entanglement, a cornerstone of quantum computation that extends beyond single-qubit manipulations, is achieved through multi-qubit gates. These gates act on two or more qubits, giving rise to interconnected quantum phenomena.

Controlled-NOT (CNOT) Gate: This gate is a two-qubit operation when the state of a target qubit is flipped if and only if the control qubit is in the $|1\rangle$ state. This entangling gate is essential in quantum circuits, enabling quantum error correction and multi-qubit logic operations. Its matrix representation is:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{11}$$

6) QUANTUM CIRCUITS

A quantum circuit is a structured arrangement of quantum gates and measurements applied to a set of qubits to execute quantum algorithms. The circuit begins with qubits initialized in a known state (commonly $|0\rangle$), followed by a sequence of quantum gates that manipulate their quantum states through unitary transformations. Formally, a quantum circuit is defined as:

$$C = (Q, G, M) \tag{12}$$

where Q represents the qubit register, G denotes the sequence of quantum gates acting on the qubits, and M signifies measurement operations, which extract classical outcomes.

A typical quantum circuit can be visually represented using circuit diagrams, as shown in Fig. 3, where horizontal lines denote qubits M and quantum gates appear as operations along these lines. The number of sequential operations defines the circuit depth N, which is a crucial factor in computational complexity. By structuring these operations efficiently, quantum circuits enable the execution of complex quantum algorithms.

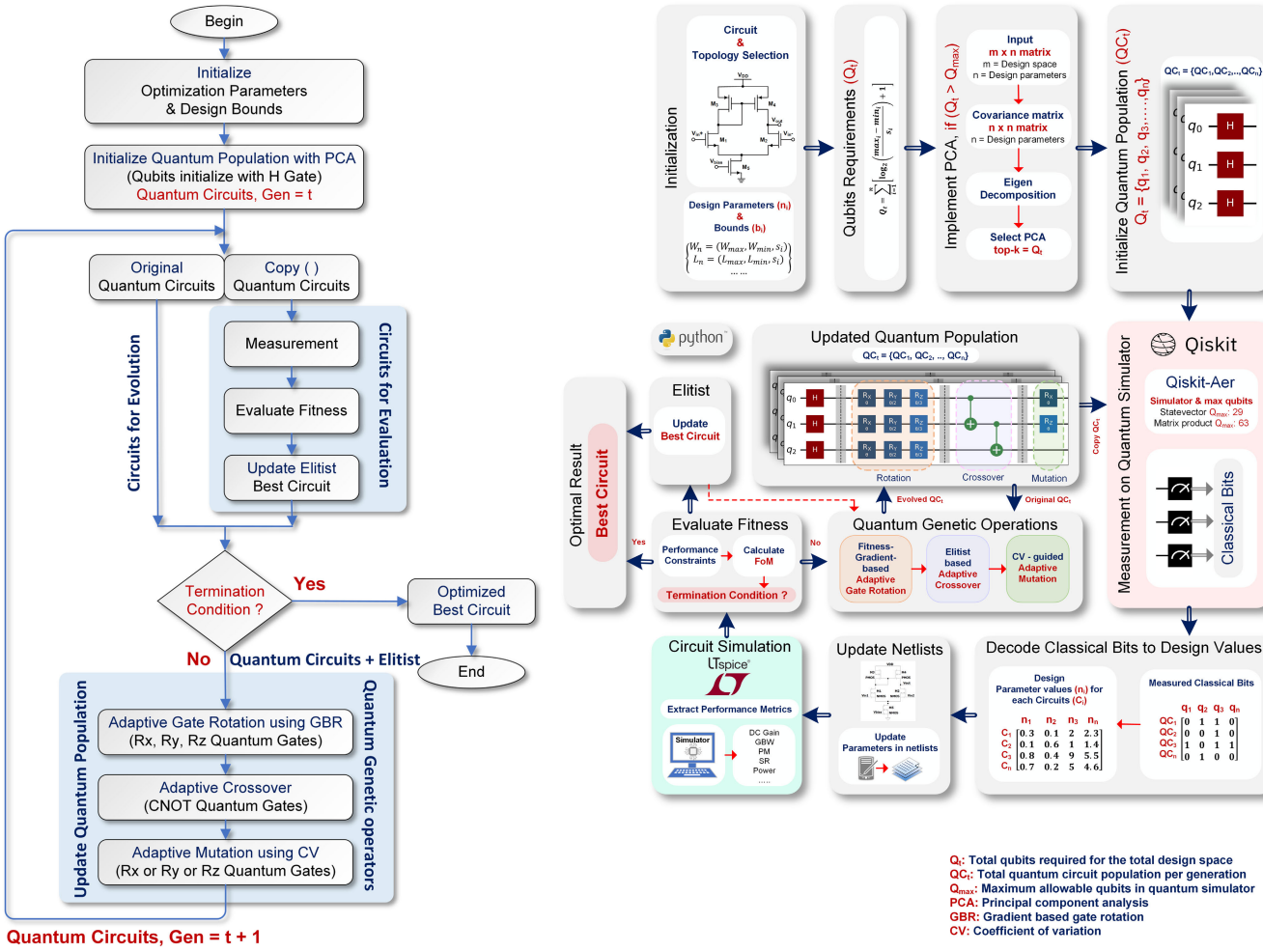


FIGURE 4. The complete systematic procedure of the EA-QGA framework.

III. ENHANCED ADAPTIVE QUANTUM GENETIC ALGORITHM

A. PROPOSED FRAMEWORK

The core concept of our Enhanced Adaptive Quantum Genetic Algorithm (EA-QGA) is to integrate quantum computing principles into the evolutionary optimization process. The proposed algorithm consists of several key stages: population generation using PCA, measurement and fitness evaluation, and enhanced quantum genetic operations. These operations include adaptive quantum gate rotation via GBR, adaptive crossover, and adaptive mutation using CV. During each generation, the best-performing circuit is continuously updated based on the evaluated fitness values. Once the termination condition–defined by the maximum number of generations–is met, the final stored best circuit represents the optimized analog circuit design parameter values. The complete operational framework of the algorithm is depicted in Fig. 4.

For implementation, we utilized Python 3.13, employing Qiskit 1.0.2 and Qiskit Aer 0.16.1, provided by IBM Quantum Platform quantum simulator. To ensure our simulations could accommodate at least 29 qubits, we focused on selecting simulators with sufficient capacity. Among the available Qiskit Aer simulators, we selected *statevector* (SV) and *matrix_product_state* (MPS) due to their specific capabilities. The SV simulator, supporting up to 29 qubits, was chosen for its comprehensive gate compatibility ($Hadamard, R_x, R_y, R_z, CNOT$) and its efficiency in debugging and analyzing intermediate quantum states. It deterministically calculates and returns the full quantum state vector, offering complete information about the quantum state at any point in the circuit. To further support the optimization process, this deterministic evolution also ensures numerically precise state information without sampling variability, which is particularly beneficial when evaluating fitness functions that depend on subtle parameter variations. This contrasts with the *qasm_simulator*, which, while also handling up to 29 qubits and mimicking real quantum hardware with measurement sampling, does not provide access to the full quantum state, offering only measurement results. For computations requiring more than 29 qubits, we transitioned to the MPS simulator.

Algorithm 1 Quantum Population Initialization With PCA Optimization

Require: B: Variable bounds, N: Population size
Ensure: $\mathcal{C}$: Quantum circuits, Q_t: Total qubits, B_s: Selected backend
  Compute required qubits Q_r for each parameter in B
  Compute total qubits: $Q_t = \sum Q_r$
  if $Q_t > 63$ **then**
    Apply PCA to reduce qubits
  end if
  Select backend B_s based on Q_t
  for $i = 1$ to N **do**
    Create quantum circuit C_i with Q_t qubits
    Apply Hadamard gates to all qubits
    Append C_i to $\mathcal{C}$
  end for

Despite potential performance degradation with excessive entanglement and limitations due to MPS approximations, it efficiently manages larger circuits with higher qubit counts, offering superior memory usage compared to other simulators handling the same qubit volume. In addition, the MPS method supports scalable simulations by adaptively maintaining tensor representations, thus allowing larger qubit systems to be evaluated while retaining high numerical stability under typical entanglement conditions encountered in our optimization circuits. Simulators like *stabilizer* and *extended_stabilizer*, while offering higher qubit capacities (10,000 and 63 respectively), were deemed unsuitable due to the stabilizer simulator's restriction to Clifford circuits and the extended_stabilizer simulator's inefficiency with deeply entangled and rotated circuits.

B. POPULATION GENERATION

Prior to initializing the quantum population, the optimization process begins by defining the analog circuit design parameters targeted for optimization. Each parameter is characterized by its operational range, specified by minimum and maximum values, and its resolution, defined by the step size. During the pre-processing phase, the number of discrete values for each parameter is calculated based on its range and resolution.

The population consists of quantum circuits, where each circuit encodes a possible analog circuit design configuration. Each chromosome represents a set of design parameters, encoded as qubits that map to discrete values within predefined bounds. In the initial stage of quantum population initialization, the EA-QGA utilizes qubit-based encoding to represent each design variable within its defined bounds. The number of qubits required for each variable is computed using logarithmic encoding, ensuring efficient representation of the discrete values within the search space. Subsequently, an appropriate quantum simulator backend B_s is selected based on the total number of qubits required. If the total number of qubits is less than or equal to 29, the SV simulator is chosen. If the total number of qubits is greater than 29 but does not exceed 63, the MPS simulator is selected. This backend selection strategy not only aligns with qubit-count feasibility but also ensures balanced accuracy and scalability, as SV provides full-state precision for smaller configurations while MPS enables larger populations to be handled efficiently with controlled resource usage.

However, when the total qubit count exceeds 63, PCA is applied to compress correlated design variables into a lower-dimensional latent space. Prior to PCA training, a synthetic dataset is generated by uniformly sampling from the discretized grids of all design variables within their respective bounds, followed by Min–Max normalization to the [0, 1] range to ensure consistent numerical scaling. PCA is then performed on this normalized dataset to identify orthogonal latent directions that capture the dominant variance of the original parameter space. The number of retained principal components (k) is determined by a twofold criterion: first, the smallest (k) is chosen such that the cumulative explained variance exceeds a predefined threshold (typically ≥ 95 %), ensuring representational fidelity; second, the discretized encoding of these (k) components is verified to remain within the available qubit budget. If the variance-based selection exceeds the hardware limit, (k) is adaptively reduced until the total qubit count fits within the simulator constraint. Each principal component is discretized with a step resolution that ensures that, after inverse transformation, the reconstructed variable values align with their original bounds and step sizes.

This PCA integration effectively removes redundant correlations among design variables while preserving the essential variance of the analog design space, ensuring that solution diversity and search-space exploration are maintained despite the dimensionality reduction. Consequently, the transformed representation enables efficient utilization of limited qubit resources without compromising the fidelity of search-space exploration. Once the qubit allocation is finalized, the EA-QGA dynamically assigns qubits to each design parameter. The subsequent stage initializes all allocated qubits into a superposition state using Hadamard gates, enabling exploration of the entire solution space simultaneously. This superposition enhances the search efficiency by leveraging quantum parallelism. The stepwise procedure for quantum population initialization with PCA optimization is presented in Algorithm 1.

C. FITNESS EVALUATION

Our algorithm utilizes a Figure of Merit (FoM), defined by Equation (1), as the fitness function. This FoM serves as the evaluation metric for candidate solutions, ensuring a consistent selection process across both quantum and classical evolutionary techniques. Following the evolutionary process in each generation, the newly populated quantum circuits are duplicated before measurement. During measurement, the duplicated circuits undergo state collapse, yielding classical data. The original circuits remain in their

quantum states, preserving quantum behavior for subsequent evolution. Decoding then translates the measured binary values into discrete circuit design parameter values within their respective bounds. To obtain reliable results, each circuit is measured over multiple shots, generating a probability distribution of outcomes. Subsequently, the extracted circuit design parameters are used to update their corresponding SPICE netlist files. These files are then executed in parallel using the LTspice circuit simulator to determine the output performance metrics for each circuit design. For each feasible candidate solution, the FoM is computed from the obtained performance metrics, and the circuit with the highest FoM is identified and stored as the elitist. This elitist is carried over to the next generation and updated whenever a new best circuit design is discovered during the evolutionary loop.

D. GRADIENT BASED ADAPTIVE GATE ROTATION

Our EA-QGA incorporates an adaptive GBR strategy that leverages fitness gradients to dynamically adjust quantum gate rotation angles (R_x, R_y, R_z), ensuring a balanced exploration and exploitation of the solution space. Fitness gradients are estimated via finite differences of successive population fitness averages, capturing the temporal evolution of performance metrics across generations. A steep positive gradient indicates regions in the design space where circuit performance improves rapidly due to favorable combinations of device dimensions and biasing conditions, whereas small or negative gradients indicates near-convergence or diminishing performance improvements. This gradient information provides directional guidance for updating qubit rotations, steering the quantum population toward superior solutions. The GBR mechanism modulates the quantum rotation angles using an adaptive scaling factor S, obtained via a sigmoid transformation of the fitness difference ΔF. This scaling factor serves as a smooth and bounded measure of performance improvement. When ΔF is large, S approaches 1, driving the base rotation angle θ_{base} toward $\theta_{\max}$ and promoting stronger exploratory behavior in high-potential design regions. Conversely, when ΔF is small or negative, S approaches 0, pulling θ_{base} toward $\theta_{\min}$ and favoring more conservative, exploitation-oriented updates. The bounds $\theta_{\min}$ and $\theta_{\max}$ are empirically set to $[0.03\pi, 0.08\pi]$. This controlled modulation ensures that changes in circuit performance translate into gradual and stable adjustments in the rotation dynamics. For implementation, the adaptation steepness parameter was set to $k = 10$ to maintain a smooth but sufficiently responsive sigmoid scaling, while the gradient amplification factor was fixed to $\lambda = 0.5$ to ensure stable yet effective modulation of the rotation updates. These values were selected empirically to provide consistent convergence behaviour across the evaluated analog circuit benchmarks.

To ensure that the finite-difference gradient is suitable for quantum-domain updates, raw gradient values are first clipped and normalized, preventing abrupt fluctuations and converting fitness-based temporal variations into a consistent directional modifier for each qubit's rotation. Temporal smoothing through momentum is then applied to further stabilize the update process across generations. Finally, the gradient is scaled to the range $[-1, 1]$, establishing a uniform bound that maintains numerical stability of the adaptive rotation factor while preserving the gradient's directionality. This ensures that evolutionary updates remain proportional and consistent across qubits, independent of population size or fitness-scale differences. This adaptive GBR approach effectively maps the physical behavior of circuit performance evolution into quantum rotational dynamics, enhancing convergence while preserving population diversity. A structured outline of this process is presented in Algorithm 2.

Algorithm 2 Adaptive Quantum Gradient-Based Rotation (GBR)

Require: $\mathcal{C}$: Quantum circuits (population), F: Fitness values, F_b: Best fitness value, t, $t_{\max}$: Current and max generations, $\theta_{\min}$, $\theta_{\max}$: Rotation angle limits, k, λ: Adaptation parameters

Ensure: Updated circuits with adaptive quantum rotations

```
Compute gradient G for F using finite differences
Normalize G to [−1, 1]
for each circuit C_i in C do
    Compute fitness difference
        ΔF = (F[i] − F[i − 1])/(|F_b|)
    Compute adaptive scaling S = 1/(1 + e^{−kΔF})
    Compute base rotation θ_base = θ_min + (θ_max − θ_min) · S
    Compute final rotation θ = θ_base · (1 + λG[i])
    Apply rotations:
        R_x(θ), R_y(θ/2), R_z(θ/3) on C_i
end for
```

E. ADAPTIVE CROSSOVER

To effectively balance exploration and exploitation, our EA-QGA utilizes an adaptive quantum crossover, designed to amplify solution diversity while safeguarding the integrity of superior individuals. This crossover mechanism leverages controlled NOT (CNOT) gates to exchange quantum states between paired circuits, ensuring effective information transfer across generations. In our implementation, adaptivity is introduced through a probabilistic selection process, where the crossover operation is applied with an initial probability set to 0.5, providing a suitable starting point for state exchange within the population. As the optimization progresses, the effect of this probability interacts with the population's fitness distribution: high-fitness individuals are increasingly exempted from crossover through elitism, while lower-fitness individuals participate in proportionally more qubit-pair exchanges through CNOT operations. This fitness-aware modulation enables the crossover operator to adjust its influence across different optimization phases without externally imposed schedules. This strategy, which preserves the elite circuit while diversifying the remaining population through selective crossover, effectively navigates the exploration–exploitation trade-off, leading to rapid

Algorithm 3 Adaptive Mutation for Quantum Circuits

Require: $\mathcal{C}$: Quantum circuits (population), F: Fitness values, P_f: Base mutation probability, ϵ = small constant
Ensure: Updated circuits with adaptive quantum mutations
  Compute mean fitness $\mu = \frac{1}{|F|}\sum F_i + \epsilon$
  Compute standard deviation $\sigma = \sqrt{\frac{1}{|F|}\sum(F_i - \mu)^2} + \epsilon$
  Compute diversity factor $CV = \sigma/\mu$
  Compute adaptive mutation rate
    $P_a = \max(0, \min(1, P_f(1 + CV)))$
  Compute maximum fitness $F_{\max} = \max(F) + \epsilon$
  for each circuit $C_i \in \mathcal{C}$ with fitness F_i **do**
    for each qubit q_j in C_i **do**
      if random value $r \sim U(0, 1) < P_a$ **then**
        Apply random gate $G \in \{R_x, R_y, R_z\}$
        with angle $\theta_j = U(0, \pi(1 - \frac{F_i}{F_{\max}}))$
      end if
    end for
  end for

convergence on optimal solutions. The quantum states of the parent chromosomes are defined by:

$$|Q_i\rangle = \bigotimes_{k=1}^{n} |\psi_{ik}\rangle, \quad |Q_j\rangle = \bigotimes_{k=1}^{n} |\psi_{jk}\rangle \tag{13}$$

where each feature k is in a superposition state as defined by equation (4).

For a single feature k, the combined state before applying the CNOT gate is:

$$|\psi_{\text{initial}}\rangle = \begin{bmatrix} \alpha_{ik}\alpha_{jk} \\ \alpha_{ik}\beta_{jk} \\ \beta_{ik}\alpha_{jk} \\ \beta_{ik}\beta_{jk} \end{bmatrix} \tag{14}$$

Applying the CNOT gate from equation (11) to the combined state $|\psi_{ik,jk}\rangle$, the final state after crossover for feature k is:

$$|\psi_{\text{final}}\rangle = \begin{bmatrix} \alpha_{ik}\alpha_{jk} \\ \alpha_{ik}\beta_{jk} \\ \beta_{ik}\beta_{jk} \\ \beta_{ik}\alpha_{jk} \end{bmatrix} \tag{15}$$

The new offspring chromosomes are formed by combining the final states for each feature:

$$|Q_{new_i}Q_{new_j}\rangle = \bigotimes_{k=1}^{n} |\psi_{\text{final}}\rangle \tag{16}$$

This operation swaps states $|10\rangle$ and $|11\rangle$ within each feature's combined state, effectively facilitating genetic information exchange while preserving quantum superposition.

F. ADAPTIVE MUTATION

Algorithm 3 defines the adaptive mutation process, which our EA-QGA employs to maintain population diversity and escape local optima. This strategy leverages the Coefficient of Variation (CV), defined as the ratio of the population's fitness standard deviation (σ) to its mean (μ), to dynamically adjust mutation intensity. Because CV reflects the spread of fitness values, it provides a direct indicator of the population's evolutionary state: a large CV (high diversity) preserves promising solutions, while a small CV (low diversity) triggers stronger mutations to explore new areas of the search space. The base mutation probability P_f is set to 0.1, providing a consistent initial mutation level aligned with the population size and problem scale. As the optimization progresses, the adaptive mutation rate P_a is modulated in response to real-time population diversity to maintain a proper balance between exploring new solutions and improving existing ones. Specifically, rotation-based quantum mutations using R_x, R_y, or R_z gates are applied to qubit states with mutation angles inversely proportional to solution fitness. This ensures weaker solutions undergo more significant probabilistic variations while stronger solutions receive only fine-grained adjustments. This adaptive approach not only introduces controlled randomness to prevent premature convergence but also enables the algorithm to escape local minima by providing larger mutations to low-fitness solutions. By balancing broad exploration with controlled mutation intensity, our adaptive mutation strategy enhances the algorithm's ability to locate the global optimum while ensuring stable and efficient convergence.

IV. PERFORMANCE EVALUATION

The effectiveness of the proposed EA-QGA algorithm was evaluated by applying it to three real-world analog circuit sizing tasks—namely, a Differential Amplifier, a Two-Stage Operational Transconductance Amplifier (Miller-OTA), and a Folded Cascode Operational Transconductance Amplifier, each systematically selected to represent increasing levels of design complexity. These circuits introduced progressively larger sets of design parameters, thereby illustrating the algorithm's robustness in exploring large, high-dimensional search spaces. All evaluations were conducted using a 45 nm CMOS process technology. For each analog circuit, the EA-QGA algorithm was executed with a population of 20 quantum individuals, each represented by a quantum circuit, over a maximum of 50 generations. This configuration reflects a balance between computational cost and convergence behavior, as real quantum-gate-based individuals and entanglement introduce significantly higher computational overhead per iteration compared to classical mimic-based approaches. Preliminary experiments showed that EA-QGA typically converges well before reaching the maximum generation limit, and increasing the population or generation count did not yield meaningful improvements in the FoM but only increased execution time. The optimization objective was guided by a power-bandwidth-oriented figure of merit (FoM), employed as the fitness function. Furthermore, the evaluation procedure was constrained by essential analog performance criteria, with particular emphasis on achieving a high open-loop DC gain, a high slew rate, a large

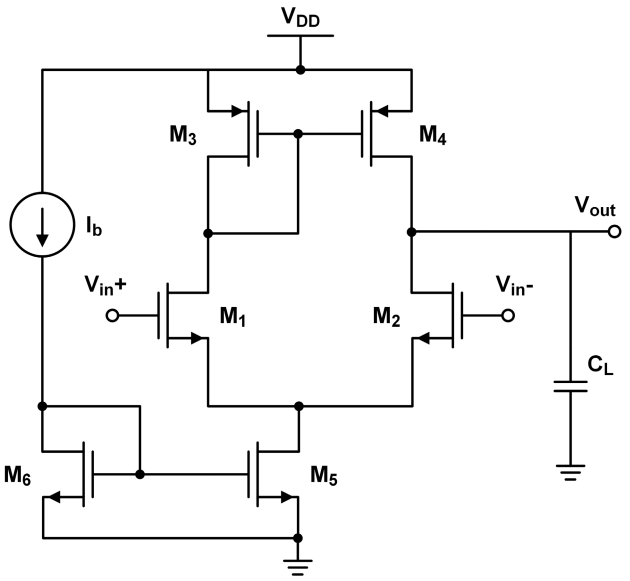


FIGURE 5. Circuit schematic of the single stage differential amplifier.

gain-bandwidth product, and a phase margin sufficient to ensure stable and reliable circuit operation. To ensure statistical reliability, 10 independent runs were performed for each sizing task, each initialized with a different random seed. Considering that hardware resource utilization directly affects overall execution time, the number of generations required to reach the optimal FoM was adopted as a practical and consistent metric for evaluating computational efficiency across all benchmark circuits. In addition to fitness-based convergence tracking, the optimization behavior was further examined by quantifying the evolution of population diversity across generations. At each generation, the spread of the design parameters across all individuals was measured, and the population diversity measure $\mathcal{S}$ was defined as the total variance of the normalized parameters, computed as the sum of the variances of each parameter across the population:

$$\mathcal{S} = \sum_{i} \mathrm{Var}(x_i) \tag{17}$$

where $\mathrm{Var}(x_i)$ denotes the variance of the i-th normalized design parameter across the population. This scalar metric characterizes the global spread of individuals in the design space and provides insight into whether the algorithm maintains sufficient exploratory capability throughout the optimization process. The results presented in the following subsections correspond to the best-performing run, identified by the highest FoM achieved with the fewest generations.

To assess the relative performance of the proposed EA-QGA approach, a comparative analysis was conducted against a quantum-inspired genetic algorithm (QIGA), wherein quantum computational behavior is emulated through classical mechanisms without the implementation of quantum gates. The QIGA was configured with a population of 50 individuals and evolved over a maximum of 150 generations. These settings were chosen to align

TABLE 1. Bound constraints for differential amplifier design variables.

Parameter	LB_i	UB_i
$W_1 - W_6$ (μm)	0.5	100
I_b (μA)	1	50
V_b (V)	0.4	0.8

TABLE 2. Tabulated results of optimized parameters for differential amplifier.

Parameter	Value	Parameter	Value
W_1, W_2 (μm)	86.2	I_b (μA)	50
W_3, W_4 (μm)	11.5	V_b (V)	0.65
W_5, W_6 (μm)	37.4		

TABLE 3. Key performance metrics for the optimized differential amplifier.

Performance	Value	Performance	Value
DC Gain (dB)	36.02	PM $(^\circ)$	86.11
GBW (MHz)	40.07	Power (μW)	99.1
SR $(V/\mu s)$	24.55	Total MOS Area (μm^2)	36.4

with the computational characteristics of QIGA, where the lower per-iteration cost of classical mimic-based quantum representations generally necessitates a larger search effort to reach competitive optima. In contrast, the reduced population and generation count selected for EA-QGA reflect its ability to exploit true quantum operations for faster convergence, as verified by the mean runtime statistics obtained over 10 independent runs. Additional comparisons were made with several previously reported results utilizing conventional genetic algorithms devoid of quantum computational enhancement. All experiments were executed on an Intel i7-1260P CPU operating at 2.10 GHz with 16 GB of RAM. The EA-QGA algorithm was implemented in Python, and the optimized design parameters produced were translated into SPICE netlists for circuit-level simulation. Performance validation was carried out using the open-source LTspice circuit simulator. To ensure statistical robustness in quantum measurement outcomes, each quantum circuit execution was repeated 1024 times (shots) to estimate the most probable measurement state.

A. DIFFERENTIAL AMPLIFIER

To initiate the evaluation of the proposed methodology, a single-stage differential amplifier topology was selected as a baseline circuit, as illustrated in Fig. 5. This configuration, comprising a minimal set of design variables, serves as a suitable benchmark for validating the optimization framework under controlled complexity. The circuit consists of 6 transistors; however, due to inherent device matching constraints and structural dependencies, the number of independent design variables is reduced to 5. These include the matched gate widths $W_1 = W_2, W_3 = W_4, W_5 = W_6$, along with the bias current I_b and the bias voltage V_b applied to the input transistors. The allowable ranges for these design

TABLE 4. Statistical Summary of EA-QGA and QIGA Over 10 independent runs for differential amplifier optimization.

Algorithm	Total Generations at Optimal FoM		Optimal FoM (MHz·pF/mW)				Runtime (min)
	Mean	SD	Mean	SD	Best	Worst	Mean
EA-QGA	20.4	2.22	803.8	4.66	809	796	6.79
QIGA	88.8	5.67	793.7	8.43	804	780	17.76

variables are presented in Table 1. To reduce the influence of channel length modulation, all transistors were designed with a channel length of 0.135 μm, corresponding to three times the minimum feature size ($L_{\min} = 0.045\ \mu$m) defined by the technology node [20].

Considering the defined bounds and resolution of each design variable, a total of 34 qubits was required for complete quantum encoding. Since this number remains well below the maximum allowable limit of 63 qubits, PCA-based dimensionality reduction was not employed in this configuration. Accordingly, the MPS backend was employed to simulate the quantum evolution. The 5 design variables, W_1, W_3, W_5, I_b and V_b, were individually mapped to [8, 8, 8, 6, 4] qubits, respectively. The quantum circuit was initialized by applying Hadamard gates to all qubits, thereby creating an equal superposition across the design space. The resulting circuit was subsequently subjected to evolutionary search via the proposed EA-QGA, The optimization formulation adopts the following objective expression:

$$\begin{aligned} &\text{max: FoM} \\ &\text{s.t. DC Gain} \geq 35\ \text{dB}, \\ &\quad \text{SR} \geq 5\ V/\mu s \\ &\quad \text{GBW} \geq 1\ MHz \\ &\quad 45^\circ \leq \text{PM} \leq 100^\circ \end{aligned} \tag{18}$$

where, FoM denotes the Figure of Merit, as formulated in (1), DC Gain represents the DC open-loop gain of the amplifier, SR denotes the slew rate, GBW refers to the gain-bandwidth product and PM represents the phase margin. Constraint handling during the evolutionary process follows the feasibility-based objective formulation introduced in (3), ensuring that only candidates satisfying all requirements in (18) are assigned a valid fitness value.

Table 2 outlines the optimized differential amplifier design variables derived from the proposed EA-QGA method, while the corresponding performance metrics are presented in Table 3. The 1 MHz GBW was set as a reference specification, while higher UGB values naturally resulted from the multi-objective optimization process. The results demonstrate that the optimized design meets all objective constraints outlined in (18), achieving an optimal trade-off among performance goals. Notably, the large widths selected for the input pair ($W_1 = W_2$) arise from the need to satisfy the DC gain, SR, and GBW constraints under the fixed channel length of (0.135 μm). With the channel length constrained, increasing device width becomes the primary means of improving (g_m), which is essential for achieving the required SR and GBW while maintaining a low overdrive voltage. As a result, the optimizer selects wider input devices not to increase DC gain alone, but to fulfill the coupled requirements imposed by the multi-objective trade-off surface, explaining their appearance in the final global optimum. The total MOS area reported in the results represents a pre-layout estimate obtained directly from the transistor geometries and is calculated as the sum of $W \times L$ for all MOS transistors in the optimized circuit. The same calculation approach is applied to the area values presented in the following subsections.

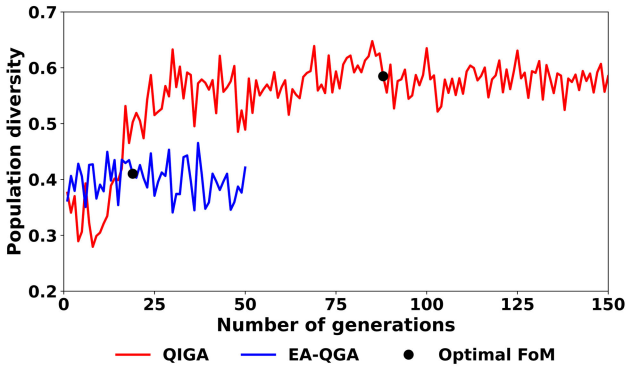


FIGURE 6. Generation-wise population diversity of EA-QGA and QIGA for the differential amplifier.

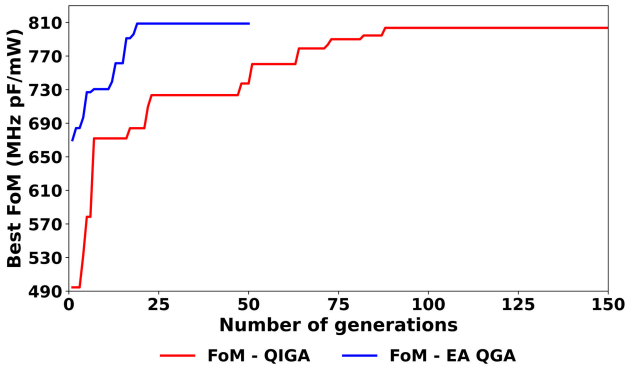


FIGURE 7. Comparison of best fitness values per generation for EA-QGA and QIGA algorithms in the design of a differential amplifier.

A statistical summary of the repeated trials is provided in Table 4 to illustrate the consistency of the proposed EA-QGA under variations in initialization. Across the 10 independent runs, the algorithm maintained a tightly grouped set of outcomes, yielding an average FoM (mean) of 803.8 with a standard deviation (SD) of 4.66 and a performance range

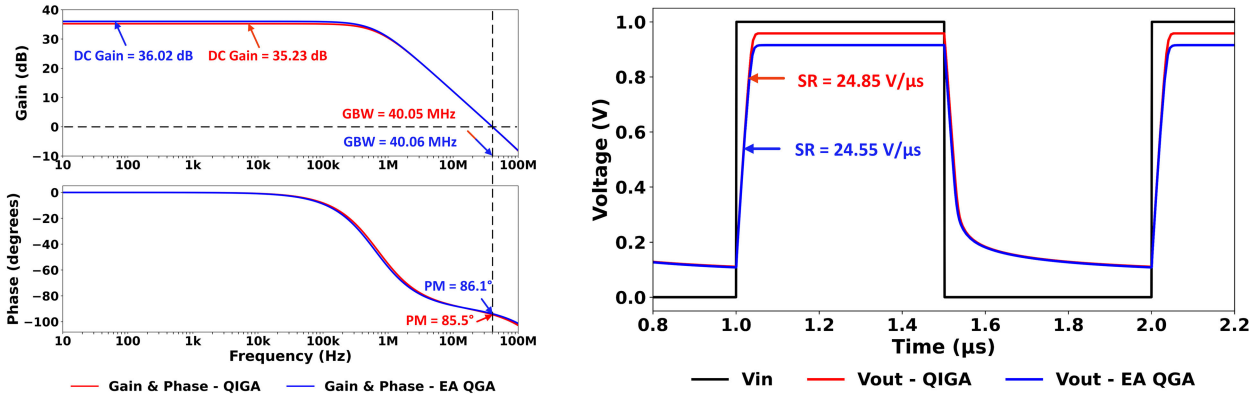


FIGURE 8. Optimized simulation outputs of differential amplifier designs using QIGA and EA-QGA methods: (a) frequency-domain response, (b) transient signal response.

spanning 809 to 796. The average number of generations required to reach the optimal FoM remained similarly concentrated around 20.4 with a deviation of 2.22, while the mean runtime stabilized at 6.79 min. These observations indicate that the optimizer exhibits reliable behavior across repeated executions, with limited dispersion in both solution quality and computational effort. The evolution of population diversity for both EA-QGA and QIGA, measured using the diversity metric $\mathcal{S}$ defined in (17), is depicted in Fig. 6. Throughout the generations, EA-QGA maintains a well-distributed population, enabling it to reach its optimal FoM in the early stages of evolution. This behavior results from diversity adjustments informed by previous-generation FoM statistics, combined with the entanglement-enabled search mechanism, which prevents premature convergence and guides the population toward coordinated updates that help identify high-quality design regions. In contrast, QIGA exhibits wider fluctuations and a gradually increasing diversity over time, indicating that it must continue adjusting its population distribution in later generations to escape less promising regions. These trends demonstrate that EA-QGA follows a more coherent and efficient search trajectory, whereas QIGA requires additional generational effort to stabilize its exploration.

As illustrated in Fig. 7, the EA-QGA framework achieved its optimal FoM of 809 at the 19th generation (out of 50) with a total execution time of approximately 6.33 min, whereas the conventional QIGA reached its slightly lower optimal FoM of 804 only at the 88th generation (out of 150), requiring a total of 17.60 min. This difference shows that QIGA needed considerably more generational iterations to approach its optimum, highlighting its slower convergence dynamics. These results indicate that EA-QGA not only reaches a better global optimum but also converges in a substantially shorter wall-clock duration—approximately 2.8× faster in elapsed time and 4.6× faster in generational convergence—demonstrating its enhanced ability to escape local optima and efficiently approach the global solution. While absolute runtime is inherently dependent on the underlying hardware configuration, the measurements included here confirm that the proposed method maintains a clear advantage in both algorithmic and practical computational efficiency. It is important to note that the higher initial FoM observed in the EA-QGA framework (670) compared to QIGA (494) in the first generation arises from differences in the quantum population generation strategies employed by the two approaches. In EA-QGA, the initial population is constructed by applying Hadamard gates across qubit groups corresponding to individual design variables, promoting a more uniform and well-distributed exploration of the design space from the outset. In contrast, QIGA relies on a probabilistic encoding of quantum states based on amplitude parameters, which, although diverse, may result in initial populations that are less likely to represent high-performing regions early in the optimization. This variation in the sampling behavior of the initial populations can naturally lead to differences in early FoM values. Hence, the observed difference in initial FoM values can be attributed to the inherent differences in the initialization strategies of the two frameworks.

Furthermore, a comparative assessment of key performance metrics, validated through frequency and transient response simulations, are illustrated in Fig. 8, further demonstrates the enhanced design efficiency of the EA-QGA approach over the conventional QIGA. Additionally, Table 5 compares the FoM values of the proposed design with those from the existing literature, confirming that the EA-QGA achieves superior performance.

B. TWO STAGE OTA

As a more intricate benchmark, a Miller-compensated two-stage operational transconductance amplifier (Miller-OTA) is employed in the second experimental case, as depicted in Fig. 9. This circuit introduces a greater degree of design complexity, involving multiple parameters such as transistor sizing, biasing conditions, and compensation

TABLE 5. Performance and FoM-based comparative evaluation of optimization methods for differential amplifier design.

Parameters	GA [14]	DE [18]	MOEA/D [18]	NSGA-II [18]	NSGA-III [18]	PB-PSO [28]	Ts-CPD [29]	LPSPSO [30]	QIGA	Proposed EA-QGA
Year	1995	2021	2021	2021	2021	2023	2023	2024	2025	2025
Process Technology (nm)	2400	180	180	180	180	45	350	350	45	45
Population Size	100	70	70	70	70	35	-	-	50	20
Total Generations	150	100	100	100	100	100	20	200	150	50
Number of Components	13	10	10	10	10	7	7	8	8	8
Supply Voltage (V)	±2.5	±0.9	±0.9	±0.9	±0.9	1.8	±2.5	±2.5	1	1
Load Capacitor, C_L (pF)	2	5	5	5	5	0.25	2.1	4	2	2
Gain (dB)	60	33	32	33	36	32.34	40.3	44.07	35.23	36.02
Gain Bandwidth (MHz)	4.8	17.72	17.68	16.21	15.82	19.3	10	13.03	40.06	40.07
Phase Margin (°)	72	89	89	85	87	92	86.1	82.36	85.5	86.11
Slew Rate (V/μsec)	3.2	10	10	10	10	8.24	24.3	17.79	24	24.55
Power Dissipation (μW)	31	832	818	782	773	8.24	1075	539.8	99.7	99.1
Total MOS Area (μm^2)	6500	1659.8	1700	1622	1615	36.7	109	247.1	42.1	36.4
FoM (MHz pF/mW)	309.67	106	108	104	102	586	19.5	96.6	804	**809**

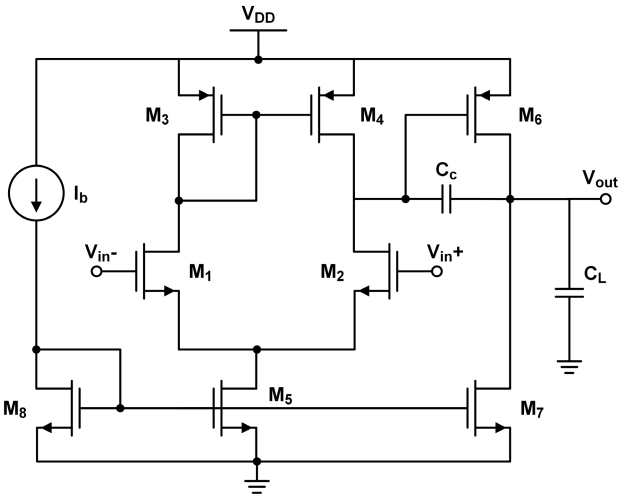


FIGURE 9. Circuit schematic of the miller compensated two-stage OTA.

TABLE 6. Bound constraints for two-stage OTA design variables.

Parameter	LB_i	UB_i
$W_1 - W_8$ (μm)	0.5	100
I_b (μA)	1	50
V_b (V)	0.4	0.8
C_c (pF)	2.2	10

elements. Although the schematic comprises 8 transistors, its symmetric structure and design constraints reduce the number of effective design variables to 8. These include the matched gate widths $W_1 = W_2$, $W_3 = W_4$, $W_5 = W_8$, along with W_6, W_7, the bias current I_b, the bias voltage V_b applied to the input stage, and the compensation capacitance C_c. The respective parameter ranges considered for each of these design variables are detailed in Table 6. Each transistor was implemented with a channel length of 0.135 μm to effectively suppress channel length modulation effects.

Considering the defined bounds and resolution of each design parameter, a total of 57 qubits was required to achieve full quantum representation. As this qubit requirement

TABLE 7. Tabulated results of optimized parameters for two-stage OTA.

Parameter	Value	Parameter	Value
W_1, W_2 (μm)	24.1	W_7 (μm)	49.5
W_3, W_4 (μm)	16.3	I_b (μA)	40
W_5, W_8 (μm)	18.8	V_b (V)	0.65
W_6 (μm)	81	C_c (pF)	2.2

TABLE 8. Key performance metrics for the optimized two-stage OTA.

Performance	Value	Performance	Value
DC Gain (dB)	68.94	PM (°)	60.7
GBW (MHz)	27.89	Power (μW)	179.9
SR ($V/\mu s$)	17.68	Total MOS Area (μm^2)	33.6

remains below the maximum threshold of 63 qubits, no PCA dimensionality reduction technique was necessary for this configuration. The simulation of quantum evolution was conducted using the MPS backend, selected for its efficiency in handling quantum systems. The 8 design variables, specifically W_1, W_3, W_5, W_6, W_7, I_b, V_b and C_c, were individually mapped to [8, 8, 8, 8, 8, 6, 4, 7] qubits, respectively, based on their resolution requirements. To ensure a uniform sampling of the search space, the quantum circuit was initialized using Hadamard gates applied across all qubits, thereby generating an equal superposition of all possible design configurations. The initialized circuit was then subjected to evolutionary optimization using the proposed EA-QGA framework. The optimization process was guided by the following objective formulation:

$$\begin{aligned} &\text{max: FoM} \\ &\text{s.t. DC Gain} \geq 60\ \text{dB}, \\ &\quad \text{SR} \geq 5\ V/\mu s \\ &\quad \text{GBW} \geq 10\ MHz \\ &\quad 60^\circ \leq \text{PM} \leq 90^\circ \end{aligned} \tag{19}$$

where, FoM denotes the Figure of Merit, as formulated in (1), DC Gain represents the DC open-loop gain of the amplifier,

TABLE 9. Statistical Summary of EA-QGA and QIGA over 10 independent runs for two stage OTA optimization.

Algorithm	Total Generations at Optimal FoM		Optimal FoM (MHz·pF/mW)				Runtime (min)
	Mean	SD	Mean	SD	Best	Worst	Mean
EA-QGA	18.4	1.89	302	5.81	310	292	7.66
QIGA	102.3	7	290	7.88	300	276	20.46

SR denotes the slew rate, GBW refers to the gain-bandwidth product and PM represents the phase margin. The feasibility rule defined in (3) governs constraint satisfaction, ensuring that only designs meeting the criteria in (19) contribute to fitness evaluation.

The optimized design variables for the two-stage OTA circuit, derived using the EA-QGA method, are summarized in Table 7, while their corresponding performance metrics are listed in Table 8. All metrics satisfy the design constraints specified in (19), ensuring functional and performance compliance. A compact statistical overview for the two-stage OTA is provided in Table 9. Across repeated executions, EA-QGA maintained tightly grouped performance outcomes, reflected in its average FoM (mean) of 302 with an SD of 5.81 and a performance range from 310 to 292. The number of generations required to reach the optimum also remained concentrated, averaging 18.4 with an SD of 1.89, while the mean runtime stabilized at 7.66 min. These results indicate that the optimizer preserves stable behavior under different initializations, with limited dispersion in both FoM and convergence effort for this benchmark.

A trend consistent with the differential-amplifier case is observed in the diversity evolution shown in Fig. 10. EA-QGA maintains a broad but well-regulated population spread during the early generations, enabling effective exploration of the enlarged two-stage OTA design space and supporting its rapid approach to the optimal FoM. In contrast, QIGA exhibits wider oscillations and a progressively expanding diversity range as it adapts to the increased search-space complexity, indicating a need for additional generations to achieve population regularity. These observations highlight the intrinsic search dynamics of both algorithms: EA-QGA organizes its exploration more coherently across generations, whereas QIGA adjusts more gradually under the larger design-space demands.

As depicted in Fig. 11, the EA-QGA method achieved a superior FoM of 310 at the 16th generation (out of 50), with a total execution time of 6.66 min, whereas QIGA reached a lower FoM of 300 only at the 93rd generation (out of 150), requiring 18.60 min. These results show that EA-QGA converged approximately 5.8× faster in generational progression and 2.8× faster in runtime while also achieving a better global optimum. While the absolute runtime inherently depends on the underlying hardware configuration, the recorded measurements clearly highlight the strengthened exploration–exploitation balance and enhanced global-search capability offered by the proposed method. Although QIGA began with a slightly higher initial FoM (212) than EA-QGA

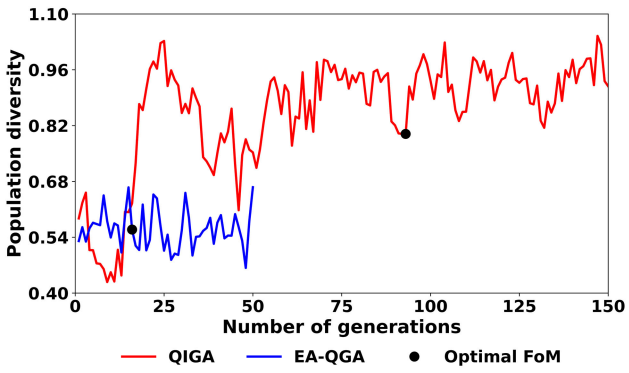


FIGURE 10. Generation-wise population diversity of EA-QGA and QIGA for the two-stage OTA.

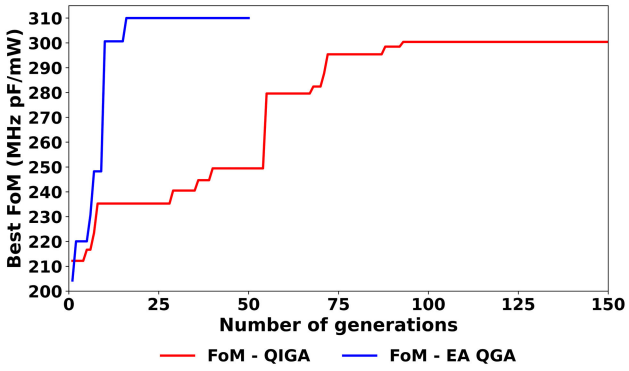
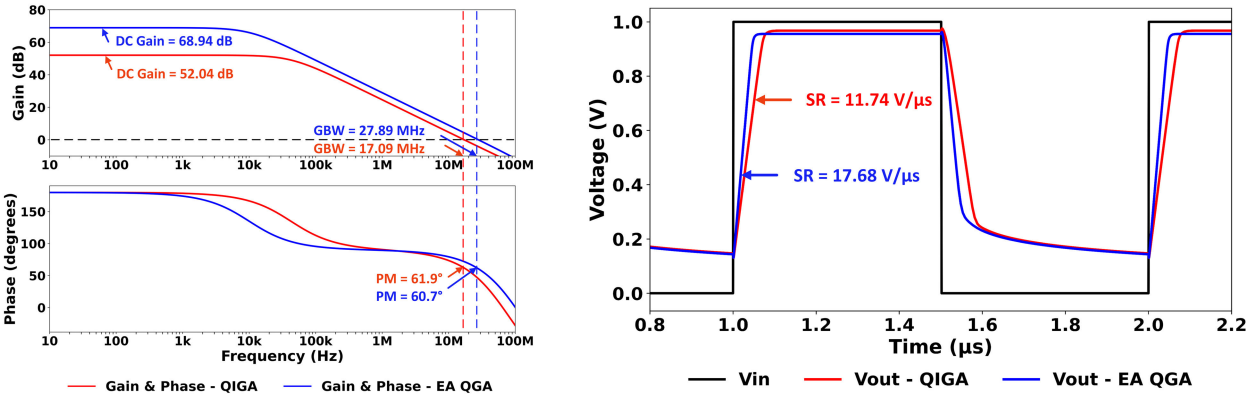


FIGURE 11. Comparison of best fitness values per generation for EA-QGA and QIGA algorithms in the design of a two-stage OTA.

(204), reflecting natural variations introduced by stochastic population sampling and encoding strategies. The adaptive capabilities of EA-QGA to rapidly surpass this initial gap and achieve superior convergence performance highlights its inherent strengths in global exploration and efficient navigation away from suboptimal regions. Further analysis of the optimized design's frequency and transient responses, illustrated in Fig. 12, confirms the improved performance achieved through EA-QGA. In addition, Table 10 compares the obtained FoM with values reported in previous studies, highlighting that EA-QGA achieves notable performance gains in two-stage OTA design.

C. FOLDED CASCODE OTA

To further evaluate the scalability and robustness of the proposed optimization framework, a folded cascode operational transconductance amplifier is employed as the third



(a) (b)

FIGURE 12. Optimized simulation outputs of two-stage OTA designs using QIGA and EA-QGA methods: (a) frequency-domain response, (b) transient signal response.

TABLE 10. Performance and FoM-based Comparative Evaluation of Optimization Methods for Two-Stage OTA Design.

Parameters	GENOM [12]	SOA [34]	PB-PSO [28]	Ts-CPD [29]	LPSPSO [30]	EB-PSO [31]	QIGA	Proposed EA-QGA
Year	2010	2022	2023	2023	2024	2024	2025	2025
Process Technology (nm)	180	350	45	350	350	45	45	45
Population Size	32	-	35	-	-	35	50	20
Total Generations	150	200	100	20	200	100	150	50
Number of Components	15	11	11	10	11	11	11	11
Supply Voltage (V)	±0.9	±2.5	1.8	±2.5	±2.5	1.8	1	1
Load Capacitor, C_L (pF)	3	4.2	1	7.1	7.2	1	2	2
Gain (dB)	70.6	78.15	60.4	64.7	78	59.88	52.04	68.94
Gain Bandwidth (MHz)	15.35	5.81	14.3	6.2	5.36	16.9	17.09	27.89
Phase Margin (°)	79.6	66.1	79.76	46	68.3	76.17	61.9	60.7
Slew Rate (V/μsec)	15.36	11.67	15.2	11.9	10.27	18.6	11.74	17.68
Power Dissipation (μW)	244	708.5	52	1084	644.1	62.43	113.8	179.9
Total MOS Area (μm^2)	1687	107.08	39.92	45.6	138.4	35.122	37.1	33.6
FoM (MHz pF/mW)	189	34.4	275	40.6	59.9	273	300	**310**

benchmark topology, as illustrated in Fig. 13. This amplifier architecture presents a more intricate design landscape, attributed to its increased number of transistors, biasing branches, and passive components. The schematic comprises a total of 15 transistors, yet due to its inherent symmetry and well-defined design constraints, the number of effective design variables is reduced to 10. These include the matched gate widths $W_1 = W_2$, W_3, $W_4 = W_5 = W_{14}$, $W_6 = W_7 = W_{13}$, $W_8 = W_9 = W_{10} = W_{11}$, W_{12}, and W_{15}, along with the resistor pair $R_1 = R_2$, the bias current I_b, and the bias voltage V_b supplied to the input differential stage. The specific design parameter ranges for each of these variables are summarized in Table 11. All transistors are implemented with a uniform channel length of 0.135 μm to minimize the impact of channel length modulation and ensure consistent performance across the design space.

For the folded cascode operational transconductance amplifier, the complete quantum encoding of all 10 design variables initially required 70 qubits, surpassing the allowable limit of 63 qubits. To address this constraint, a PCA-based dimensionality reduction strategy was adopted. Among the original design variables, comprising transistor widths $[W_1, W_3, W_4, W_6, W_8, W_{12}, W_{15}]$, resistor R_1, bias current I_b and bias voltage V_b, the 7 transistor width parameters with redundant bounds were compressed into four principal components using PCA. Consequently, the dimensionality of the design space was reduced to four effective variables: the PCA-encoded width components, R_1, I_b and V_b. These were subsequently mapped to [44, 4, 6, 4] qubits, respectively. The total number of qubits required after compression was thereby reduced to 58, which conforms to the quantum simulator constraints. The MPS backend was employed to simulate the quantum evolution of this compressed design space. As with the other test cases, the quantum circuit was initialized by applying Hadamard gates to all qubits, thereby generating an equal superposition state. The evolutionary search was then executed using the proposed EA-QGA approach, which iteratively evolved quantum states toward optimal design configurations. The optimization process was governed by

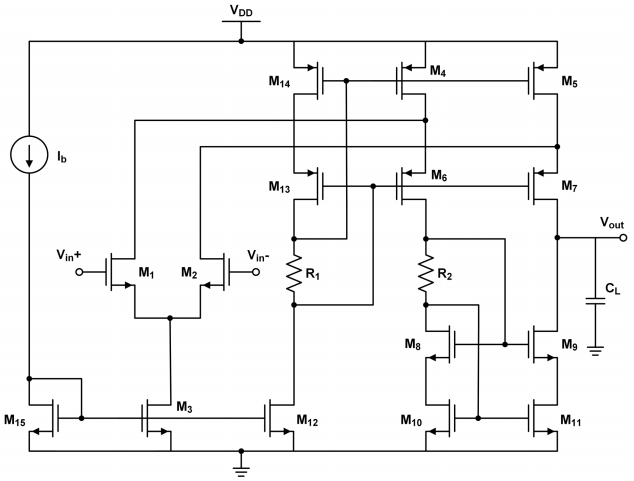


FIGURE 13. Circuit schematic of the folded cascode OTA.

TABLE 11. Bound constraints for folded cascode OTA design variables.

Parameter	LB_i	UB_i
$W_1 - W_{15}$ (μm)	0.5	100
R_1, R_2 $(k\Omega)$	1	10
I_b (μA)	1	50
V_b (V)	0.4	0.8

TABLE 12. Tabulated results of optimized parameters for folded cascode OTA.

Parameter	Value	Parameter	Value
W_1, W_2 (μm)	53.7	W_{12} (μm)	62.3
W_3 (μm)	95.5	W_{15} (μm)	24
W_4, W_5, W_{14} (μm)	30	R_1, R_2 $(k\Omega)$	9
W_6, W_7, W_{13} (μm)	15.6	I_b (μA)	20
W_8, W_9, W_{10}, W_{11} (μm)	11.3	V_b (V)	0.75

the following objective expression:

$$\begin{aligned} &\text{max: FoM} \\ &\text{s.t. DC Gain} \geq 60 \text{ dB}, \\ &\quad \text{SR} \geq 10\ V/\mu s \\ &\quad \text{GBW} \geq 10\ MHz \\ &\quad 60^\circ \leq \text{PM} \leq 90^\circ \end{aligned} \tag{20}$$

where, FoM denotes the Figure of Merit, as formulated in (1), DC Gain represents the DC open-loop gain of the amplifier, SR denotes the slew rate, GBW refers to the gain-bandwidth product and PM represents the phase margin. Constraint compliance is maintained using the formulation in (3), whereby candidates violating any requirement in (20) are automatically penalized.

Table 12 provides the optimized design parameters obtained for the folded cascode OTA circuit using the EA-QGA method, with associated performance metrics reported in Table 13. All performance outcomes comply with the constraints specified in (20), indicating a robust optimization result. The statistical results for the Folded

TABLE 13. Key performance metrics for the optimized folded cascode OTA.

Performance	Value	Performance	Value
DC Gain (dB)	68	PM $(^\circ)$	67.7
GBW (MHz)	57.35	Power (μW)	166.7
SR $(V/\mu s)$	24	Total MOS Area (μm^2)	63.6

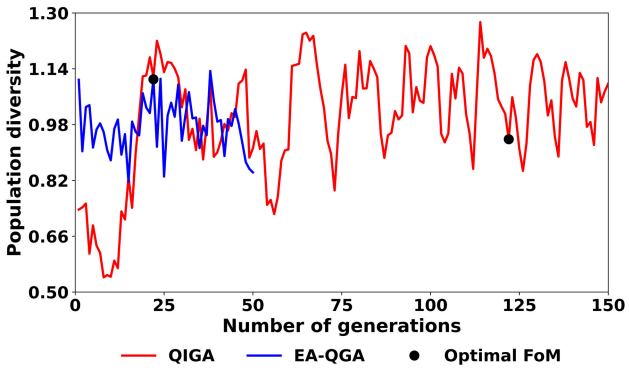


FIGURE 14. Generation-wise population diversity of EA-QGA and QIGA for the folded cascode OTA.

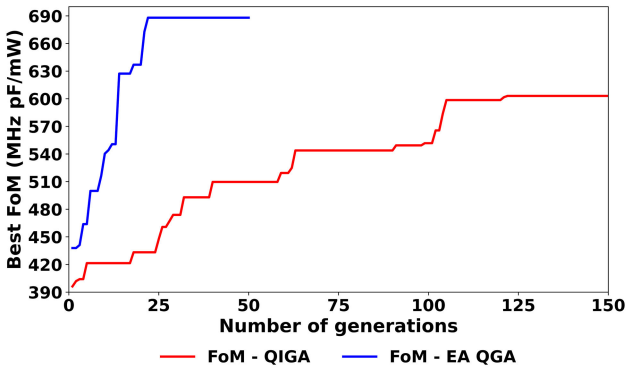
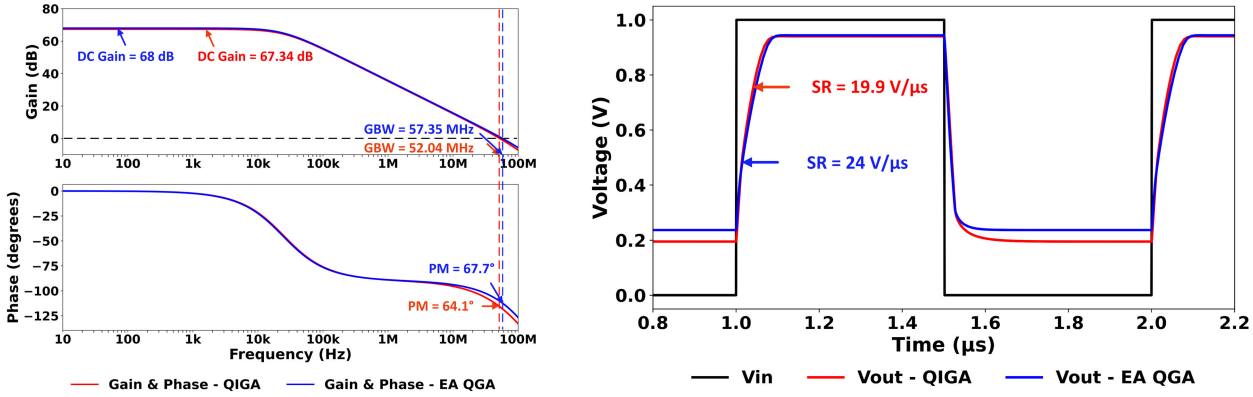


FIGURE 15. Comparison of best fitness values per generation for EA-QGA and QIGA algorithms in the design of a folded cascode OTA.

Cascode OTA, summarized in Table 14, show a similarly consistent trend. EA-QGA achieved an average FoM (mean) of 675.6 with an SD of 8.93 and a performance range spanning 688 to 661 across the 10 runs. The number of generations required to reach the optimum remained centered around 26.2 with a deviation of 4.84, while the mean runtime stabilized at 12.22 min. These observations confirm that the algorithm maintains steady performance despite variations in initialization, yielding closely clustered outcomes for this topology as well.

The population-diversity evolution in Fig. 14 shows that EA-QGA maintains a compact and relatively stable diversity range across generations, indicating a consistently regulated exploration pattern even in this more complex design scenario. This steady behavior supports its ability to progress toward high-quality regions without requiring large redistributions of its population. In contrast, QIGA exhibits pronounced and irregular shifts in diversity, with



(a) (b)

FIGURE 16. Optimized simulation outputs of folded cascode OTA designs using QIGA and EA-QGA methods: (a) frequency-domain response, (b) transient signal response.

TABLE 14. Statistical summary of EA-QGA and QIGA Over 10 independent runs for folded cascode OTA optimization.

Algorithm	Total Generations at Optimal FoM		Optimal FoM (MHz·pF/mW)				Runtime (min)
	Mean	SD	Mean	SD	Best	Worst	Mean
EA-QGA	26.2	4.84	675.6	8.93	688	661	12.22
QIGA	119.7	4.47	598.7	3.71	603	592	23.94

TABLE 15. Performance and FoM-based comparative evaluation of optimization methods for folded cascode OTA design.

Parameters	Ts-CPD [29]	MGA [13]	MGA-MLSCP [22]	QIGA	Proposed EA-QGA
Year	2023	2023	2024	2025	2025
Process Technology (nm)	350	180	180	45	45
Population Size	-	30	20	50	20
Total Generations	20	200	200	150	50
Number of Components	18	14	14	19	19
Supply Voltage (V)	±2.5	1.8	1.8	1	1
Load Capacitor, C_L (pF)	10	5	5	2	2
Gain (dB)	74.1	40.79	41.01	67.34	68
Gain Bandwidth (MHz)	17.8	40	40	52.04	57.35
Phase Margin (°)	83.9	89.88	89.89	64.1	67.7
Slew Rate (V/μsec)	13.8	21.14	21.53	19.9	24
Power Dissipation (μW)	3300	1225	1230	172.6	166.7
Total MOS Area (μm^2)	600.9	8.019	8.020	86.3	63.6
FoM (MHz pF/mW)	53.9	163	163	603	**688**

rapid rises and collapses that reflect frequent resets of its search direction. These fluctuations show that QIGA relies on broader population adjustments to recover from less productive regions of the landscape. Overall, the observed trends confirm that EA-QGA preserves a more dependable and well-balanced exploration dynamic, while QIGA follows a noticeably more volatile evolutionary path. According to the generational FoM trend illustrated in Fig. 15, EA-QGA reached its optimal FoM of 688 at the 22nd generation (out of 50), with a measured execution time of 10.26 min. In contrast, QIGA attained a lower FoM of 603 only at the 122nd generation (out of 150), requiring 24.40 min to converge. This demonstrates that EA-QGA reached a superior solution approximately 5.6× faster in generations and 2.4× faster in runtime. These improvements can be attributed to the adaptive evolutionary–quantum mechanisms in EA-QGA, which enhance global exploration and accelerate convergence across complex search spaces. A comparative evaluation of the circuit's key performance metrics, based on frequency and transient simulations and shown in Fig. 16, further supports the superior design efficiency achieved through EA-QGA. Lastly, Table 15 presents a comparative analysis of the proposed design's FoM with values reported in prior literature, clearly establishing the EA-QGA framework as a high-performing alternative for folded cascode OTA optimization.

V. CONCLUSION

This study introduced an adaptive quantum genetic algorithm-based methodology for optimizing the sizing of CMOS analog circuits. A novel EA-QGA framework was proposed to effectively address the challenge of mapping a large analog design search space onto the limited qubit resources of quantum simulators. The framework incorporates PCA-based qubit encoding along with adaptive strategies such as GBR and CV-guided mutation, which collectively enhance the exploration capability and improve the likelihood of reaching true global optima without getting trapped in a local minima. The effectiveness of the proposed approach was demonstrated through its application to the design of a differential amplifier, a two-stage OTA, and a folded cascode OTA using 45 nm CMOS technology. Across all case studies, the EA-QGA consistently satisfied the required performance constraints and surpassed the QIGA baseline by achieving FoM improvements of 0.62%, 3.33%, and 14.10%, while also converging 4.6×, 5.8×, and 5.6× faster in generations and reducing runtime by 2.8×, 2.8×, and 2.4×, respectively. In addition to these performance gains, the EA-QGA achieved notable reductions in total MOS area relative to the QIGA. Across all three benchmark circuits, the proposed framework achieved area reductions of 13.5%, 9.4%, and 26.3%, respectively, demonstrating its effectiveness in producing more compact transistor-level designs. This performance advantage was achieved even with a reduced population size and fewer generational iterations, highlighting the efficiency gained through the use of real quantum-gate-based operators and their accelerated convergence behavior. Overall, these results confirm that the EA-QGA method offers improved computational efficiency, higher optimization accuracy, and more area-efficient analog circuit implementations. Furthermore, due to its scalability and adaptability, the proposed approach holds significant potential for extension to the synthesis of more complex analog circuits involving larger and more intricate design spaces.

REFERENCES

[1] J. Scheible, "Optimized is not always optimal—The dilemma of analog design automation," in *Proc. Int. Symp. Phys. Design*, Apr. 2022, pp. 151–158, doi: 10.1145/3505170.3511042.

[2] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*. Hoboken, NJ, USA: Wiley, 2024.

[3] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 12, pp. 1247–1266, Dec. 1989, doi: 10.1109/43.44506.

[4] A. A. Youssef, B. Murmann, and H. Omran, "Analog IC design using precomputed lookup tables: Challenges and solutions," *IEEE Access*, vol. 8, pp. 134640–134652, 2020, doi: 10.1109/ACCESS.2020.3010875.

[5] E. Hjalmarson, R. Hagglund, and L. Wanhammar, "An equation-based optimization approach for analog circuit design," in *Proc. Int. Symp. Signals, Circuits Syst.*, vol. 1, Jul. 2003, pp. 77–80, doi: 10.1109/SCS.2003.1226952.

[6] Z. Zhao, T. Liao, and L. Zhang, "Fast performance evaluation for analog circuit synthesis frameworks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351560.

[7] M. D. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 1, pp. 1–21, Jan. 2001, doi: 10.1109/43.905671.

[8] S. R. Chowdhury, S. Bhardwaj, and J. Kitchen, "Design automation of CMOS op-amps using statistical geometric programming," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2022, pp. 1575–1579, doi: 10.1109/ISCAS48785.2022.9937871.

[9] J. Li, H. Zhi, X. Jiang, K. Zhu, and Y. Li, "Decoupling analog circuit representation from technology for behavior-centric optimization," in *Proc. 62nd ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2025, pp. 1–7, doi: 10.1109/DAC63849.2025.11133189.

[10] S. Hong, Y. Tae, D. Lee, G. Park, J. Lim, K. Cho, C. Jeong, M.-J. Park, S. Hong, and J. Han, "Analog circuit design automation via sequential RL agents and Gm/ID methodology," *IEEE Access*, vol. 12, pp. 104473–104489, 2024, doi: 10.1109/ACCESS.2024.3435331.

[11] J. Li, H. Zhi, R. Lyu, W. Li, Z. Bi, K. Zhu, Y. Zeng, W. Shan, C. Yan, F. Yang, Y. Li, and X. Zeng, "AnalogGym: An open and practical testing suite for analog circuit synthesis," 2024, *arXiv:2409.08534*.

[12] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration*, vol. 43, no. 1, pp. 136–155, Jan. 2010, doi: 10.1016/j.vlsi.2009.09.001.

[13] R. Rashid, G. Raghunath, V. Badugu, and N. Nambath, "Performance evaluation of evolutionary algorithms for analog integrated circuit design optimisation," *Microelectron. J.*, vol. 141, Nov. 2023, Art. no. 105983, doi: 10.1016/j.mejo.2023.105983.

[14] W. Kruiskamp and D. Leenaerts, "Darwin: Analogue circuit synthesis based on genetic algorithms," *Int. J. Circuit Theory Appl.*, vol. 23, no. 4, pp. 285–296, Jul. 1995, doi: 10.1002/cta.4490230404.

[15] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137–148, Feb. 2009, doi: 10.1016/j.vlsi.2008.04.003.

[16] M. Barari, H. R. Karimi, and F. Razaghian, "Analog circuit design optimization based on evolutionary algorithms," *Math. Problems Eng.*, vol. 2014, no. 1, pp. 1–12, Jan. 2014, doi: 10.1155/2014/593684.

[17] M. Shokouhifar and A. Jalali, "An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1189–1201, Feb. 2015, doi: 10.1016/j.eswa.2014.09.030.

[18] M. A. Valencia-Ponce, E. Tlelo-Cuautle, and L. G. de la Fraga, "On the sizing of CMOS operational amplifiers by applying many-objective optimization algorithms," *Electronics*, vol. 10, no. 24, p. 3148, Dec. 2021, doi: 10.3390/electronics10243148.

[19] J. Li, H. Zhi, W. Shan, Y. Li, Y. Zeng, and Y. Li, "Multi-task evolutionary to PVT knowledge transfer for analog integrated circuit optimization," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2023, pp. 1–9, doi: 10.1109/ICCAD57390.2023.10323899.

[20] A. Lberni, M. A. Marktani, A. Ahaitouf, and A. Ahaitouf, "Analog circuit sizing based on evolutionary algorithms and deep learning," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121480, doi: 10.1016/j.eswa.2023.121480.

[21] M. Campilho-Gomes, R. Tavares, and J. Goes, "Analog flat-level circuit synthesis with genetic algorithms," *IEEE Access*, vol. 12, pp. 115532–115545, 2024, doi: 10.1109/ACCESS.2024.3446308.

[22] R. Rashid, K. Krishna, C. P. George, and N. Nambath, "Machine learning driven global optimisation framework for analog circuit design," *Microelectron. J.*, vol. 151, Sep. 2024, Art. no. 106362, doi: 10.1016/j.mejo.2024.106362.

[23] Z. Xu, Z. Zhao, and J. Liu, "Deterministic multi-objective optimization of analog circuits," *Electronics*, vol. 13, no. 13, p. 2510, Jun. 2024, doi: 10.3390/electronics13132510.

[24] J. Li, Y. Zeng, H. Zhi, J. Yang, W. Shan, Y. Li, and Y. Li, "Knowledge transfer framework for PVT robustness in analog integrated circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 5, pp. 2017–2030, May 2024, doi: 10.1109/TCSI.2023.3340683.

[25] J. Li, H. Zhi, X. Jiang, Y. Zeng, W. Shan, and Y. Li, "Balancing objective optimization and constraint satisfaction for robust analog circuit optimization," in *Proc. Asia South Pacific Design Autom. Conf.*, 2025, pp. 190–196, doi: 10.1145/3658617.3697701.

[26] C. Tang, X. Chen, Y. Luo, and Y. Zeng, "Automatic optimal design method for circuit sizing based on CNN surrogate model assisted differential evolution algorithm," *IEEE Access*, vol. 12, pp. 136238–136247, 2024, doi: 10.1109/ACCESS.2024.3462952.

[27] Y. Ben and G. Shi, "Applying design equations in particle swarm optimization for auto-sizing of multi-stage opamps: An experimental study," *Anal. Integr. Circuits Signal Process.*, vol. 103, no. 1, pp. 117–130, Apr. 2020, doi: 10.1007/s10470-019-01555-2.

[28] 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. 123577–123588, 2023, doi: 10.1109/ACCESS.2023.3329698.

[29] P. Lagos-Eulogio, P. Miranda-Romagnoli, J. C. Seck-Tuoh-Mora, and N. Hernández-Romero, "Improvement in sizing constrained analog IC via ts-CPD algorithm," *Computation*, vol. 11, no. 11, p. 230, Nov. 2023, doi: 10.3390/computation11110230.

[30] R. Das, B. P. De, S. Ghosh, R. Kar, D. Mandal, and B. Appasani, "Optimal CMOS analog amplifier circuit design using a modified PSO for consumer electronic applications," *J. Electr. Comput. Eng.*, vol. 2024, no. 1, pp. 1–14, Jan. 2024, doi: 10.1155/jece/2004118.

[31] K. G. Shreeharsha, R. K. Siddharth, M. H. Vasantha, and Y. B. N. Kumar, "An error bound particle swarm optimization for analog circuit sizing," *IEEE Access*, vol. 12, pp. 50126–50136, 2024, doi: 10.1109/ACCESS.2024.3385491.

[32] A. Fortes, L. A. Da Silva Jr., R. Domanski, and A. Girardi, "Two-stage OTA sizing optimization using bio-inspired algorithms," *J. Integr. Circuits Syst.*, vol. 14, no. 3, pp. 1–10, Dec. 2019, doi: 10.29292/jics.v14i3.74.

[33] G. I. Tombak, S. N. Güzelhan, E. Afacan, and G. Dündar, "Simulated annealing assisted NSGA-III-based multi-objective analog IC sizing tool," *Integration*, vol. 85, pp. 48–56, Jul. 2022, doi: 10.1016/j.vlsi.2022.02.009.

[34] K. B. Maji, B. P. De, R. Kar, D. Mandal, and S. P. Ghoshal, "CMOS analog amplifier circuits design using seeker optimization algorithm," *IETE J. Res.*, vol. 68, no. 2, pp. 1376–1385, Mar. 2022, doi: 10.1080/03772063.2019.1649207.

[35] K. Touloupas and P. P. Sotiriadis, "LoCoMOBO: A local constrained multiobjective Bayesian optimization for analog circuit sizing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 9, pp. 2780–2793, Sep. 2022, doi: 10.1109/TCAD.2021.3121263.

[36] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 61–66, doi: 10.1109/ICEC.1996.542334.

[37] J. Ur Rehman, M. S. Ulum, A. W. Shaffar, A. A. Hakim, Mujirin, Z. Abdullah, H. Al-Hraishawi, S. Chatzinotas, and H. Shin, "Evolutionary algorithms and quantum computing: Recent advances, opportunities, and challenges," *IEEE Access*, vol. 13, pp. 16649–16670, 2025, doi: 10.1109/ACCESS.2025.3530952.

[38] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982, doi: 10.1007/bf02650179.

[39] M. A. Nielsen, and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[40] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002, doi: 10.1109/TEVC.2002.804320.

[41] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," *Proc. Congr. Evol. Comput.*, vol. 2, pp. 1354–1360, Jul. 2002, doi: 10.1109/CEC.2000.870809.

[42] Y. Rubio, C. Olvera, and O. Montiel, "Quantum-inspired evolutionary algorithms on IBM quantum experience," *Eng. Lett.*, vol. 29, no. 4, p. 1573, 2021.

[43] S. S. Rosales-Alvarado, O. Montiel, U. Orozco-Rosas, and J. J. Tapia, "Developing a quantum genetic algorithm in MATLAB using a quantum device on AWS," in *New Directions on Hybrid Intelligent Systems Based on Neural Networks, Fuzzy Logic, and Optimization Algorithms*, vol. 1146. Cham, Switzerland: Springer, 2024, pp. 111–127, doi: 10.1007/978-3-031-53713-4_10.

[44] S. S. Rosales, O. Montiel, U. Orozco-Rosas, J. J. Tapia, and O. Castillo, "Comparison of performance of Amazon braket using a quantum genetic algorithm," *Computación Y Sistemas*, vol. 28, no. 3, pp. 1429–1448, Sep. 2024, doi: 10.13053/cys-28-3-5178.

[45] S. Altares-López, J. J. García-Ripoll, and A. Ribeiro, "AutoQML: Automatic generation and training of robust quantum-inspired classifiers by using evolutionary algorithms on grayscale images," *Expert Syst. Appl.*, vol. 244, Jun. 2024, Art. no. 122984, doi: 10.1016/j.eswa.2023.122984.

[46] H. Wang, J. Liu, J. Zhi, and C. Fu, "The improvement of quantum genetic algorithm and its application on function optimization," *Math. Problems Eng.*, vol. 2013, pp. 1–10, 2013, doi: 10.1155/2013/730749.

[47] A. A. Abdulhussien, M. F. Nasrudin, S. M. Darwish, and Z. A. A. Alyasseri, "Feature selection method based on quantum inspired genetic algorithm for Arabic signature verification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 3, pp. 141–156, Mar. 2023, doi: 10.1016/j.jksuci.2023.02.005.

[48] H. M. H. Saad, R. K. Chakrabortty, S. Elsayed, and M. J. Ryan, "Quantum-inspired genetic algorithm for resource-constrained project-scheduling," *IEEE Access*, vol. 9, pp. 38488–38502, 2021, doi: 10.1109/ACCESS.2021.3062790.

[49] A. G. Pai, K. M. Buddhiraju, and S. S. Durbha, "Quantum inspired genetic algorithm for bi-level thresholding of gray-scale images," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. XLVIII-4/W6-2022, pp. 483–488, Feb. 2023, doi: 10.5194/isprs-archives-xlviii-4-w6-2022-483-2023.

[50] A. S. Hesar and M. Houshmand, "A memetic quantum-inspired genetic algorithm based on Tabu search," *Evol. Intell.*, vol. 17, no. 3, pp. 1837–1853, Jun. 2024, doi: 10.1007/s12065-023-00866-8.

[51] J. Wen, X. Qu, J. Liu, S. Lin, and Q. Xiao, "A novel fault location method for the active distribution network based on dynamic quantum genetic algorithm," *Electr. Eng.*, vol. 106, no. 4, pp. 4719–4735, Aug. 2024, doi: 10.1007/s00202-024-02244-8.

[52] T. H. Pham and B. Raahemi, "An adaptive quantum-based genetic algorithm feature selection for outlier detection," in *Proc. 10th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, vol. 267, Jul. 2024, pp. 1034–1039, doi: 10.1109/codit62066.2024.10708347.

P. SRIDHAR received the B.E. degree in electronics and communication engineering and the M.E. degree in mechatronics engineering from Anna University, India, in 2011. He is currently pursuing the Ph.D. degree with Vellore Institute of Technology, India.

From 2011 to 2022, he was an Assistant Professor with the Faculty of Electronics and Communication Engineering, Anna University-affiliated Institute. His research interests include analog and mixed-signal IC design, evolutionary computation, machine learning, graph neural networks, and quantum computing applied to EDA optimization.

HARISH MALLIKARJUN KITTUR (Senior Member, IEEE) was born in Gadag, Karnataka, India. He received the B.Sc. degree in physics, mathematics, and electronics from Karnataka University, Dharwad, India, the M.Sc. degree in physics from Indian Institute of Technology (IIT) Bombay, Mumbai, India, the M.Tech. degree in solid state technology from IIT Madras, Chennai, India, and the Ph.D. degree from the 2nd Physics Institute, RWTH Aachen University, Germany.

He was with the Department of Electronics and Communication Engineering, SDM College of Engineering and Technology (SDMCET), Dharwad, India. From 2012 to 2015, he was the Division Leader of the VLSI Division and the Program Manager for the M.Tech. VLSI Program with VIT, Vellore. From 2019 to 2022, he was the Dean of the School of Electronics Engineering, VIT, Vellore. He is currently a Professor with the School of Electronics Engineering, Vellore Institute of Technology (VIT), Vellore, India. He has focused on electronics engineering education and research, with over 23 years of experience in this field. His current research interests include semiconductor physics, devices and circuits, integrated circuit design, magnetoelectronics, nanoelectronics, and memory design.

Prof. Kittur is a regular reviewer of the *International Journal of Electronics*, and has also served as a reviewer for several other journals. He has been a member of several Boards of Studies. He was a recipient of the DAAD Fellowship for his M.Tech. project work, at RWTH Aachen University, in Germany. He was awarded the M.Tech. Institute Merit Prize by IIT Madras.