

# Automated Analog Circuit Design by Evolutionary Algorithms

Sureshbhai L Bharvad<sup>1</sup>, Pankajkumar P Prajapati<sup>2\*</sup>

<sup>1</sup>Gujarat Technological University, Ahmedabad, Gujarat, India, <sup>2</sup>Vishwakarma Government Engineering College, Chandkheda, Gujarat, India. \*Corresponding Author's Email: pankaj@ldce.ac.in

## Abstract

Mixed-mode integrated circuits become increasingly popular in modern VLSI Chip design. However, designing these circuits in CMOS technology is no small feat given the need to balance a range of conflicting performance criteria and design constraints. In this work, optimized two-stage operational amplifier designed using meta heuristics ABC (Artificial Bee Colony) and PSO (Particle Swarm Optimization) algorithms. By leveraging ABC and PSO algorithms, the design process can efficiently achieve the desired performance goals by optimally sizing of the circuit components. In the proposed automated optimization environment, implementation of the ABC and PSO algorithms are implemented in Python language and integrated with the Ngspice simulation tool using BSIM4 MOSFET models in TSMC's 130nm technology node. The entire optimization environment is set up on the Ubuntu operating system, and the results achieved with the ABC and PSO are compared with earlier reported work in which proposed optimized operational amplifier achieved higher unity gain bandwidth and higher CMRR. Proposed automated design environment interface between Ngspice circuit simulator and evolutionary algorithm to obtain the optimum results for the mentioned multidimensional design problem.

**Keywords:** ABC Algorithm, Automated Environment, Operational Amplifier Design, Optimization, PSO Algorithm.

## Introduction

Analog circuits are very important in electronics because all modern VLSI chips have analog and digital blocks on the same chip. This makes designing analog circuits, such as op-amps a quite interesting and challenging task. Although there are many optimization tools for digital circuits to improve their performance, there is also a growing need for automated design tool for analog circuit is required. In analog design after choosing proper circuit topology, the next step is to set the right values for the circuit components so that the circuit meets its target specifications. However, as physical models become more complex and technology scales down, figuring out these values by hand calculations becomes very difficult task. This is why efficient, automated design methods are needed. Optimizing an electronic circuit involves selecting appropriate component values to ensure the desired performance. To streamline this process, numerous advanced optimization techniques have been developed for automated analog circuit design.

Finding solution from Gradient based methods requires calculation of derivatives and optimal initial guess of the design parameters (1). If

starting point isn't close to the overall best solution, these methods tend to find only a local optimum. Steepest descent search (2), Sequential Quadratic Programming Method (3), Levenberg-Marquardt (4), are the examples of gradient based methods. Convex optimization method gives global optimal solution but require a deep understanding of the circuit design and accurate device models which makes modern design more challenging (5). Evolutionary algorithms offer another solution by effectively handling problems with multiple optima. They sidestep many issues faced by convex and gradient-based methods because they do not depend on detailed internal physics MOSFET device, and they avoid difficult mathematical computations while still finding the global optimum (6). Genetic Algorithms have been widely adopted by many researchers as an effective optimization technique for various complex problems. Several studies have demonstrated the applicability of GAs in optimizing analog circuit designs shows its ability to solve the nonlinear and multimodal problems (7). The PSO algorithm exhibits superior computational efficiency and consistency compar-

This is an Open Access article distributed under the terms of the Creative Commons Attribution CC BY license (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

(Received 01<sup>st</sup> July 2025; Accepted 06<sup>th</sup> October 2025; Published 30<sup>th</sup> October 2025)

ed to the GA in addressing transistor sizing problems, thereby makes it effective alternative to replace GA (8). The authors concluded that the PSO algorithm effectively meets the required design specifications while minimizing the overall MOS transistor area. The ABC algorithm exhibits considerable exploration capability but exploitation ability remains relatively limited (9). Due to this limitation, the ABC algorithm tends to exhibit slower convergence on unimodal problems and trapped in local minima for complex multimodal optimization tasks. Recent research has increasingly focused on AI-driven approaches with neural network assisted circuit synthesis, reducing the reliance on time consuming SPICE simulations (10). Reinforcement learning has been applied to sequential circuit construction and sizing, enabling agents to learn efficient design strategies through interaction with the simulator. Likewise, Bayesian optimization offers a probabilistic, surrogate based framework that balances exploration and exploitation, making it well suited for high dimensional and computationally expensive design problems (11). These AI techniques complement traditional evolutionary algorithms by offering enhanced scalability, faster convergence, and the potential for knowledge transfer. Integrating such approaches into automated analog design frameworks represents a promising avenue for future research, especially as circuit architectures continue to increase in complexity.

This paper proposes an automated simulation framework for optimizing CMOS operational amplifier parameters using ABC and PSO algorithms, implemented in Python and interfaced with Ngspice circuit simulator. ABC and PSO are tested with the various unimodal and multi modal benchmark functions to evaluate the overall performance and the comparative study has been carried out. The framework is applied to a two stage CMOS operational amplifier, and the results demonstrate improved optimization efficiency

and optimized design specification compared with the previous studies.

### The Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm models the foraging behavior of bees to find food in a search space (12). The algorithm operates through several key stages. Initially, food sources are generated randomly to create the starting population. In the employed bee stage, each bee explores the local search space to improve its assigned food source, and information about the food source is communicated to neighboring bees using a greedy selection mechanism. During the onlooker bee stage, food sources with higher probability of improvement are selected and further exploited. In the scout bee stage, food sources that have not shown improvement are abandoned, and new sources are randomly chosen from the search space. In this way, employed bees focus on improving all food sources, while onlooker bees concentrate on those with the highest likelihood of enhancement (13).

### The Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is a nature-inspired method that mimics the flocking behavior of birds. In PSO, each bird in the flock is modeled as a particle that explores the problem space with its own position and velocity (14). These particles update their velocities based on both their individual past performance and the successes of neighboring particles, gradually steering toward the best-found solution over successive iterations. This iterative process continues till optimum results are not obtained. Let,  $N$  is swarm number and  $D$  is dimension of swarm. The velocity and position of the  $n^{th}$  particle is updated by  $V_n = [V_{n1}, V_{n2}, \dots, V_{nD}]$  and  $x_n = [x_{n1}, x_{n2}, \dots, x_{nD}]$ . The  $n^{th}$  particle velocity updated by every each iteration is calculated by mathematical expression as stated in past study (15).

$$= \omega * V_{nd}^t + C_1 * rand_1 * (pbest_{nd}^t - x_{nd}^t) + C_2 * rand_2 * (gbest_d^t - x_{kd}^t) \quad [1]$$

Here,  $d$  is in the range of  $\{1, 2, 3, \dots, D\}$ ,  $n$  is in the range of  $\{1, 2, 3, \dots, N\}$ ,  $t$  is in the range of  $\{1, 2, 3, \dots, \text{maximum number of iteration}\}$ .  $V_{nd}^t, x_{nd}^t$  are the velocity and position of  $n^{th}$  particle respectively,  $pbest_{nd}^t$  is personal best position,  $gbest_d^t$  is global best of the  $n^{th}$  particle for the  $t^{th}$  iteration in  $d^{th}$  dimension. Random numbers  $rand_1$  and  $rand_2$  are uniformly generated between 0 and 1,  $C_1$  and  $C_2$  are the constants are also called acceleration coefficients, initially inertia weight  $\omega$  selected less than one and reduced linearly after every iteration.

The  $n^{th}$  particle position is updated by equation following equation.

$$= x_{nd}^t + V_{nd}^{t+1} \quad [2]$$

The linearly varying inertia weight  $\omega$  is calculated using mathematical equation from past study (16).

$$= (\omega_{max} - \omega_{min}) * \frac{(t_{max}-t)}{t} + \omega_{max} \quad [3]$$

Where,  $\omega_{max}$  and  $\omega_{min}$  are the maxiimum and minimum value of  $\omega$ . Current and maximum iterations are  $t$  and  $t_{max}$  respectively.

### Testing of Algorithms

To evaluate the effectiveness of the evolutionary algorithms, standard benchmark functions are employed for performance comparison (17). This study demonstrates the comparative effectiveness of well-established evolutionary algorithms, Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC), in achieving faster convergence for global optimization problems (18). The main focus of this work is on analyzing convergence behavior and robustness when applied to both unimodal and multimodal benchmark functions, which represent challenging test cases in analog

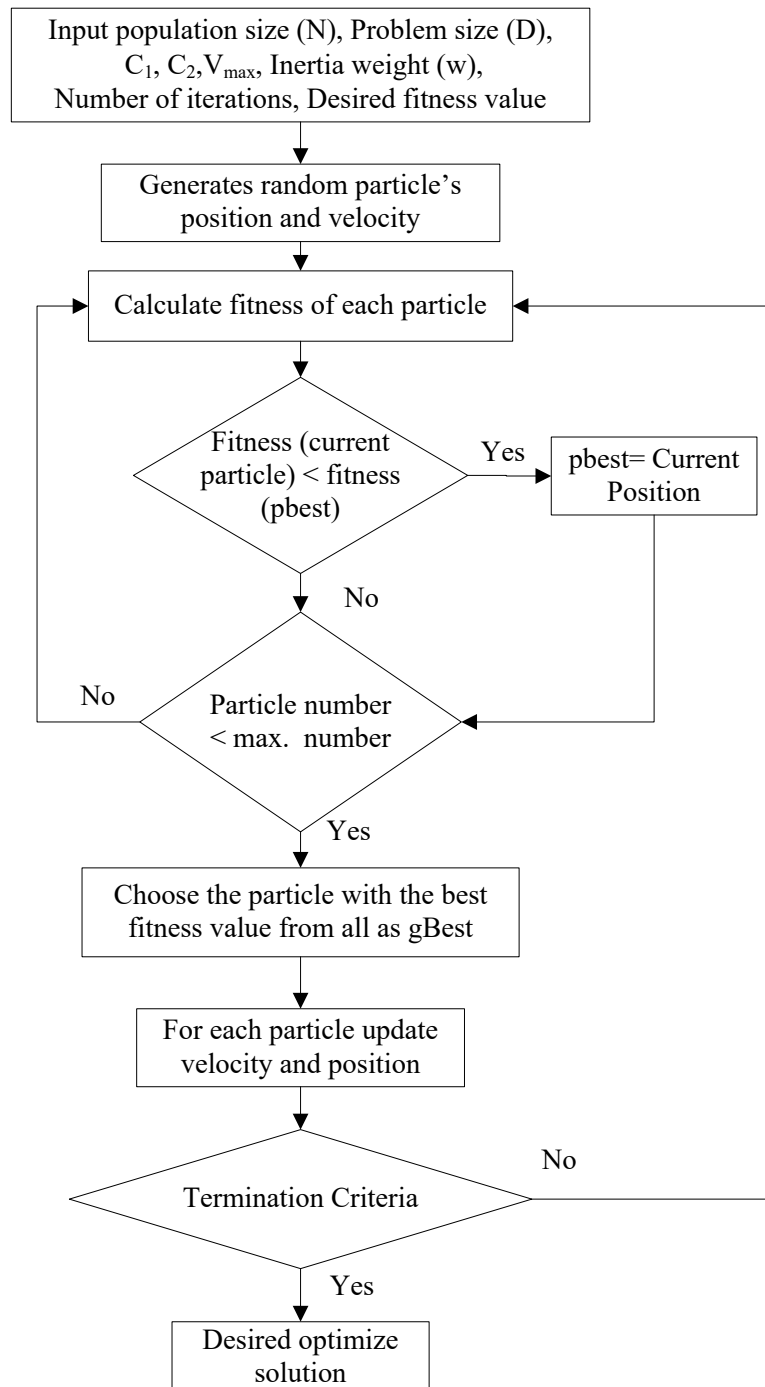
circuit optimization which is a multi-dimensional design problem. The obtained results from each algorithm are compared with the known optima of the benchmark functions to validate their efficiency in global search. A set of commonly used unimodal and multimodal functions are considered within the specified search space, as summarized in Table 1. This table lists the mathematical formulations of five standard benchmark functions used for testing optimization algorithms, along with their types (unimodal or multimodal), search spaces, and global optima.

**Table 1:** Standard Benchmark Functions for the Comparative Analysis of Metaheuristic Algorithms

| Benchmark Function | Mathematical Equation of Benchmark Function                                                                      | Type of Function | Search Space  | Global Optimal Solution |
|--------------------|------------------------------------------------------------------------------------------------------------------|------------------|---------------|-------------------------|
| Ackley             | $f_1(x) = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)} + 20 + e$ | Multi-modal      | (-32, 32)     | (0, ..., 0)             |
| Beale              | $f_2(x) = (1.5 - x_0 + x_0x_1)^2 + (2.25 - x_0 + x_0x_1^2)^2 + (2.625 - x_0 + x_0x_1^3)^2$                       | Uni-modal        | (-4.5, 4.5)   | (3, 0.5)                |
| Griewank           | $f_3(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$           | Multi-modal      | (-600, 600)   | (0, ..., 0)             |
| Rastrigin          | $f_4(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$                                                        | Multi-modal      | (-5.12, 5.12) | (0, ..., 0)             |
| Sphere             | $f_5(x) = \sum_{i=1}^D x_i^2$                                                                                    | Uni-Modal        | (-100, 100)   | (0, ..., 0)             |

For the PSO algorithm (Figure 1), it is essential to properly initialize the control parameters (C1, C2, and  $\omega$ ), population size, and maximum iterations to ensure reliable convergence. In this study, the parameter settings were adopted in line with established literature to allow fair and reproducible comparison: Number of particles =

30, C1 = 1.4962, C2 = 1.4962,  $\omega$  initially set to 0.995 and linearly decreased at each iteration. The desired fitness value is set to 1e-6, with a maximum of 1000 iterations. Termination occurs when either the fitness threshold is met or the iteration limit is reached (19).



**Figure 1:** Flow Diagram of the PSO Algorithm (6)

**Table 2:** Control Parameters of Algorithms

| Sr. No. | Parameters of ABC algorithm   | Parameters of PSO algorithm  |
|---------|-------------------------------|------------------------------|
| 1       | Total Number of Bees (N) = 30 | Number of Particles (N) = 30 |
| 2       | Fitness = 1e-6                | Fitness = 1e-6               |
| 3       | Maximum Iteration = 1000      | Maximum Iteration = 1000     |
| 4       | Dimension = 10                | Dimension = 10               |
| 5       | Number of Runs = 100          | Number of Runs = 100         |

For the uniformity the majority of the parameters of ABC and PSO algorithms are set equal and listed in Table 2 which lists common control parameters

like population size, dimension, and maximum iterations used in the ABC and PSO algorithms.

**Table 3:** Standard Benchmark Function Solution Results

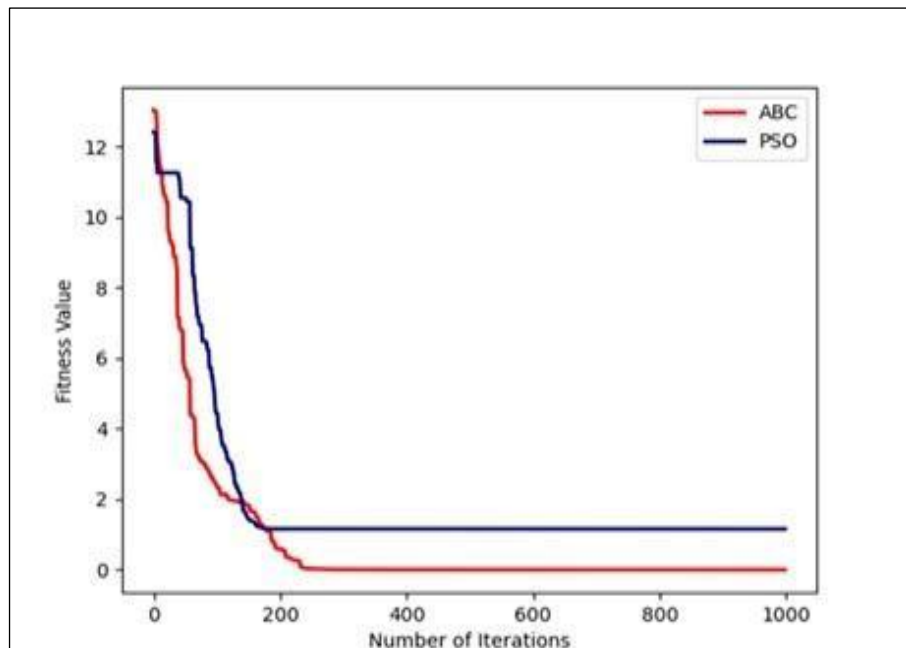
| Benchmark functions |                      | ABC     | PSO     |
|---------------------|----------------------|---------|---------|
| Ackley              | Min                  | 2.32e-7 | 1.72e-7 |
|                     | Max                  | 9.99e-7 | 9.96e-7 |
|                     | Mean                 | 8.20e-7 | 6.51e-7 |
|                     | SD                   | 1.59e-7 | 2.33e-7 |
|                     | Total Time (S)       | 39.9    | 5.34    |
|                     | Average Iteration    | 354     | 129     |
|                     | Confidence Intervals | 100     | 100     |
| Beale               | Min                  | 4.72e-8 | 3.15e-9 |
|                     | Max                  | 7.14e-5 | 7.62e-1 |
|                     | Mean                 | 4.37e-6 | 3.04e-2 |
|                     | SD                   | 1.26e-5 | 1.50e-1 |
|                     | Total Time (S)       | 31.41   | 3.75    |
|                     | Average Iteration    | 599     | 121     |
|                     | Confidence Intervals | 75      | 96      |
| Griewank            | Min                  | 8.48e-2 | 1.47e-2 |
|                     | Max                  | 5.14e-1 | 4.20e-1 |
|                     | Mean                 | 3.08e-1 | 9.85e-2 |
|                     | SD                   | 8.42e-2 | 6.24e-2 |
|                     | Total Time (S)       | 106.96  | 96.2    |
|                     | Average Iteration    | 1000    | 1000    |
|                     | Confidence Intervals | 0       | 0       |
| Rastrigin           | Min                  | 4.09    | 1.98    |
|                     | Max                  | 33.69   | 37.80   |
|                     | Mean                 | 20.92   | 10.88   |
|                     | SD                   | 5.84    | 6.16    |
|                     | Total Time (S)       | 40.67   | 56.14   |
|                     | Average Iteration    | 1000    | 1000    |
|                     | Confidence Intervals | 0       | 0       |
| Sphere              | Min                  | 8.31e-8 | 2.60e-7 |
|                     | Max                  | 9.96e-7 | 9.94e-7 |
|                     | Mean                 | 7.01e-7 | 7.37e-7 |
|                     | SD                   | 2.38e-7 | 1.85e-7 |
|                     | Total Time (S)       | 13.63   | 5.09    |
|                     | Average Iteration    | 184     | 150     |
|                     | Confidence Intervals | 100     | 100     |

Final results of benchmark functions for ABC and PSO algorithms are compared and solutions obtained for this standard benchmark functions are listed in Table 3, where metrics such as minimum value, mean, standard deviation, iteration count, and confidence intervals are reported. The corresponding convergence plots for these benchmark functions are shown in Figure 2 to Figure 6. As shown in Figure 2 convergence behavior of ABC and PSO for the Ackley function which highlights PSO's faster optimization speed. In the Figure 3, convergence

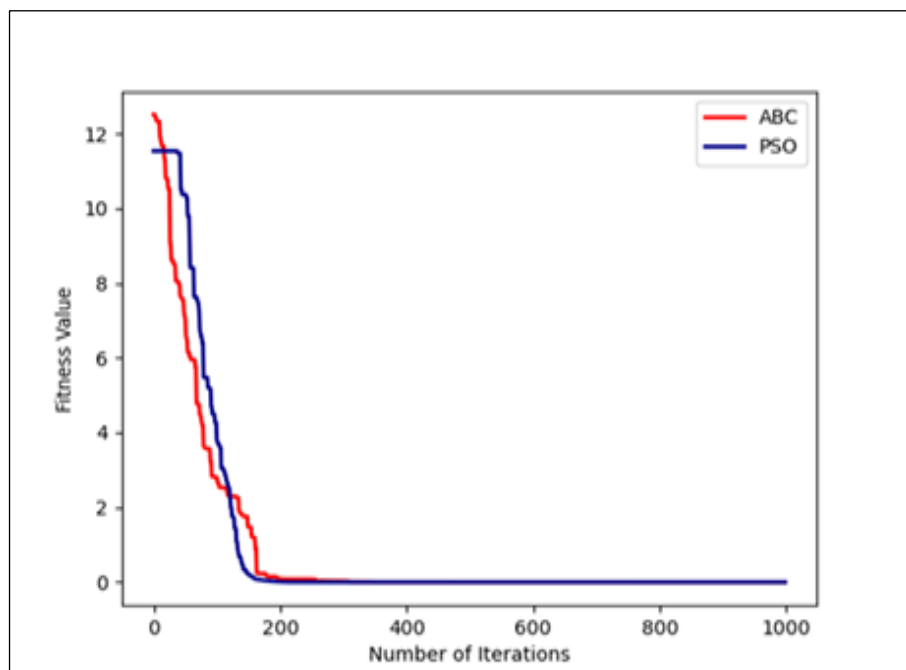
plot for the Beale function demonstrates that PSO rapidly achieves the global minimum, while ABC converges more gradually. Figure 4 convergence plot for the Griewank function, illustrates convergence of both algorithms on a multimodal function, shows ABC's robustness against local optima. Figure 5, shows convergence plot for the Rastrigin function where PSO converges faster but ABC is more stable. Figure 6, Highlights PSO's superior convergence speed on a simple unimodal function compared to ABC. These plots clearly illustrate the faster convergence behavior of PSO

compared to ABC, particularly for unimodal benchmark functions where PSO quickly reaches the global optimum. In contrast, ABC shows more gradual convergence but demonstrates greater stability in multimodal functions, avoiding premature convergence to local optima. The

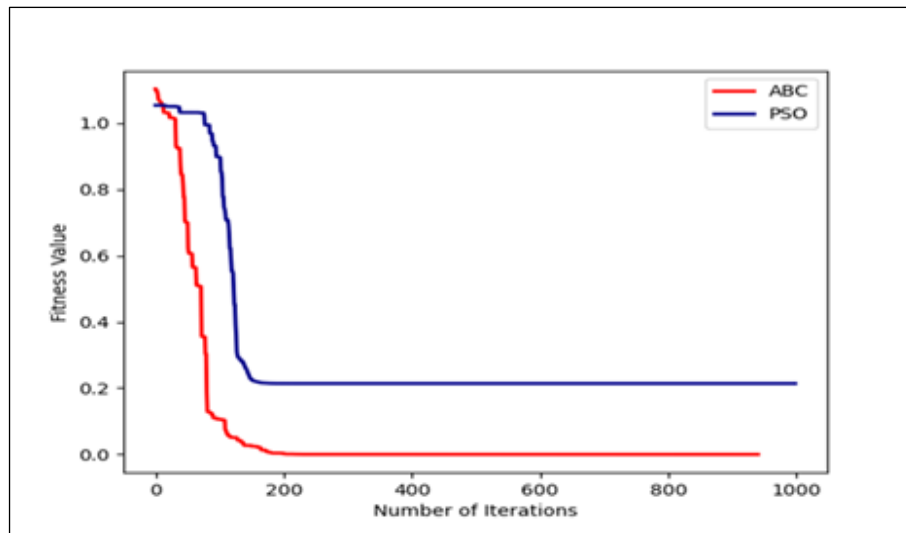
quantitative outcomes summarized that PSO achieves higher accuracy and lower error margins across simpler functions, while ABC provides more consistent results as the problem complexity and dimensionality increase.



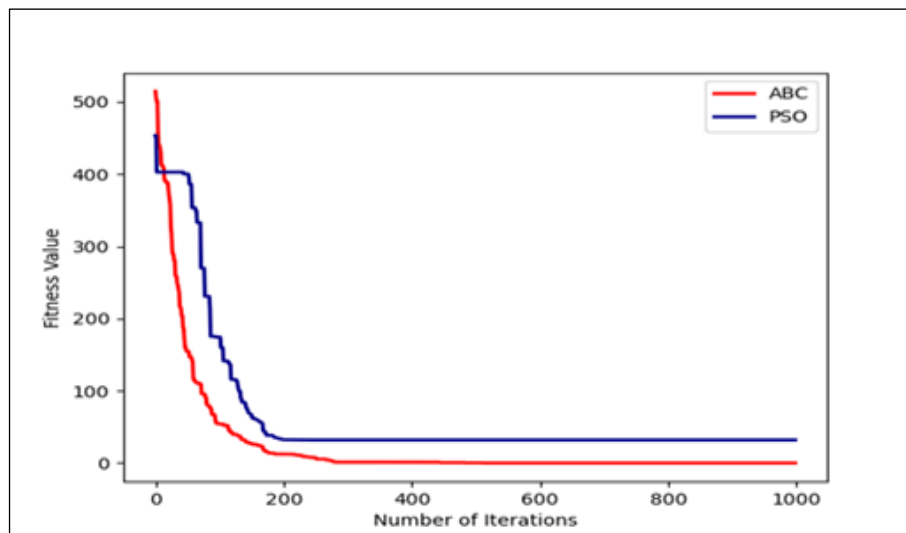
**Figure 2:** Convergence Plot for the Ackley Function



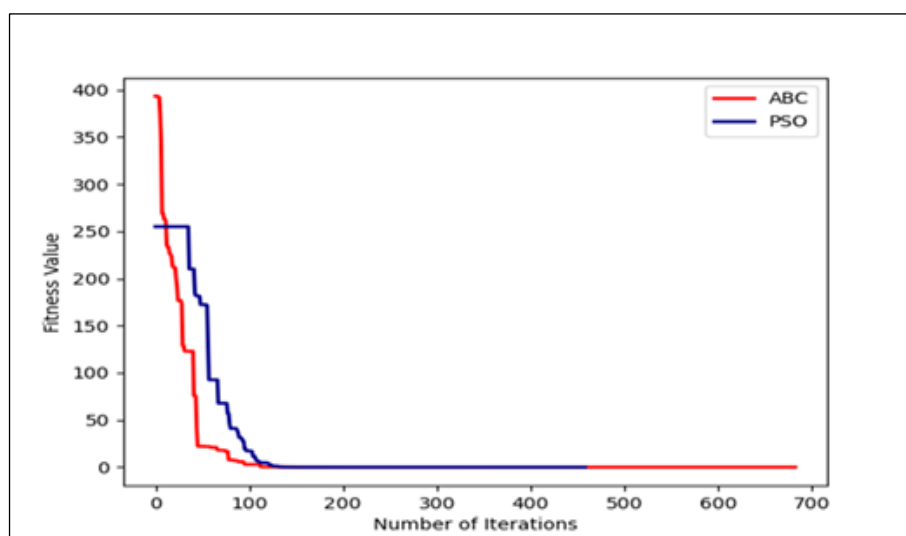
**Figure 3:** Convergence Plot for the Beale Function



**Figure 4:** Convergence Plot for the Griewank Function



**Figure 5:** Convergence Plot for the Rastrigin Function



**Figure 6:** Convergence Plot for the Sphere Function

## Two Stage Operational Amplifier

The operational amplifier is widely used as a fundamental building block in analog CMOS-based systems, including integrators, differentiators, summers, oscillators, amplifiers, and filters. It can be considered the backbone of analog circuits. The two-stage op-amp is particularly popular due to its advantages, such as high input impedance and low output impedance. In CMOS-based two-stage op-amps, the input stage typically consists of a differential pair, followed by an additional gain stage at the output and associated biasing circuits, as illustrated in Figure 7.

### Input Differential Gain Stage

The differential gain stage is made up by two NMOS transistors having width  $W_1$  and length  $L_1$  and input is given from  $V_{in1}$  and  $V_{in2}$  to the gate of the transistors. Where,  $V_{in1}$  and  $V_{in2}$  acts as the inverting and non-inverting input respectively. The two PMOS having width  $w_2$  and length  $L_2$  work as active load for the differential amplifier which provides higher output resistance and improves the common-mode rejection ratio (CMRR) (20).

### Second Gain Stage

The additional gain is provided by the PMOS transistor having width  $W_3$  and length  $L_2$  connected in common source configuration. This PMOS transistor gets the amplified input from the first differential gain stage and further amplified by this second gain stage. The NMOS transistor

having width  $W_4$  and length  $L_3$  act as active load resistor for above common source output stage amplifier. The PMOS transistor with width  $W_5$  and length  $L_4$  provides connection between input differential stage and output gain stage. Thus this stage provides additional gain to the operation amplifier in addition to the first stage (21).

### Biasing Circuit

The cascode current mirror is formed by the two PMOS transistors having widths  $W_7$ ,  $W_8$  and length  $L_5$  and circuit's biasing is handled by these transistors. The two NMOS transistor having widths  $W_6$ ,  $W_9$  and length  $L_3$  act as an active mirror current source (22). The NMOS transistor having width  $W_9$  and length  $L_3$  also act as an active load transistor. The optimize value of load capacitor is selected to 0.5 pF. Power supply  $V_{DD}$  is set to  $\pm 0.6V$  for the 130 nm technology node. The overall gain depends on the transconductance of the differential stage transistors.

## Methodology

In CMOS based analog circuit design, it is important to keep specific relationships between circuit parameters, like the channel length  $L$  and channel width  $W$  of the transistors, to obtain the required specifications. Figure 8 shows an automated optimization environment for analog based CMOS circuits, which is part of a larger tool where the circuit topology is already chosen by the user.

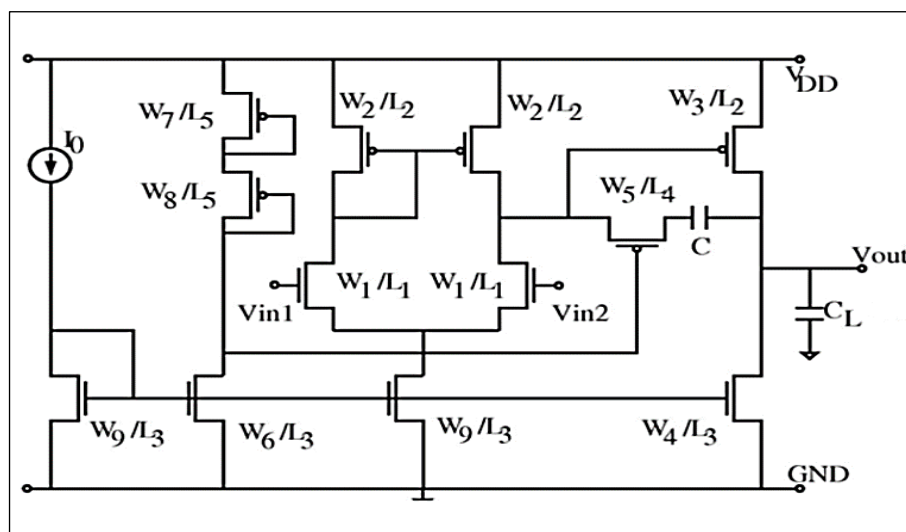


Figure 7: Two Stage Operational Amplifier

## Proposed Automated Optimization Environment

This design system has two main parts: an optimizer and a SPICE simulation environment. The optimizer uses evolutionary algorithms (EA) such as ABC and PSO to adjust circuit parameters until the desired performance is reached. Each design produced by the EA is checked with the Ngspice simulator to make sure it works correctly (23). A SPICE netlist for each circuit is created to test its performance against the required targets. It is important that the transistors work in the saturation region, so the netlist is set up with the

proper voltage transfer characteristic (VTC) and bias points.

The software interface manages the communication between the optimizer and the simulator. A script provides the necessary coordination by accepting inputs like the EA settings, the circuit design parameters, and the specifications to be met. Initially, the EA randomly picks values for the design parameters within defined limits, and these values are then sent to the simulator (24). In this equation, After the simulation is run, a fitness function is calculated using the following equation:

$$= \sqrt{\sum_{j=1}^D \left( \frac{Specification_{desired} - Specification_{simulated}}{Specification_{desired}} \right)^2_j} \quad [4]$$

D is the total number of specifications. This root-mean-square (RMS) error formula provides equal weight to each specification. The optimizer's main goal is to lower the fitness function value with each iteration until it reaches an acceptable level. The process stops when either the fitness function value drops below a set minimum 1e-6 or reaching to maximum number of iterations. If the stopping criteria aren't met, the optimization algorithms generate a new set of design parameters and the cycle repeats (25). The design targets listed in Table 4 (gain, phase margin, UGB, PSRR, CMRR,

and slew rates) constitute a multi-objective optimization problem (26). These specifications are often conflicting and must be balanced rather than optimized independently (27). In the present work we map the multi-objective task onto a single objective by using a normalized root-mean-square (RMS) error fitness metric as shown in above equation. This simplifies algorithm comparison and allows direct evaluation of convergence behavior and robustness for ABC and PSO under identical conditions.

**Table 4:** Design Specification of Two Stage Op-Amp

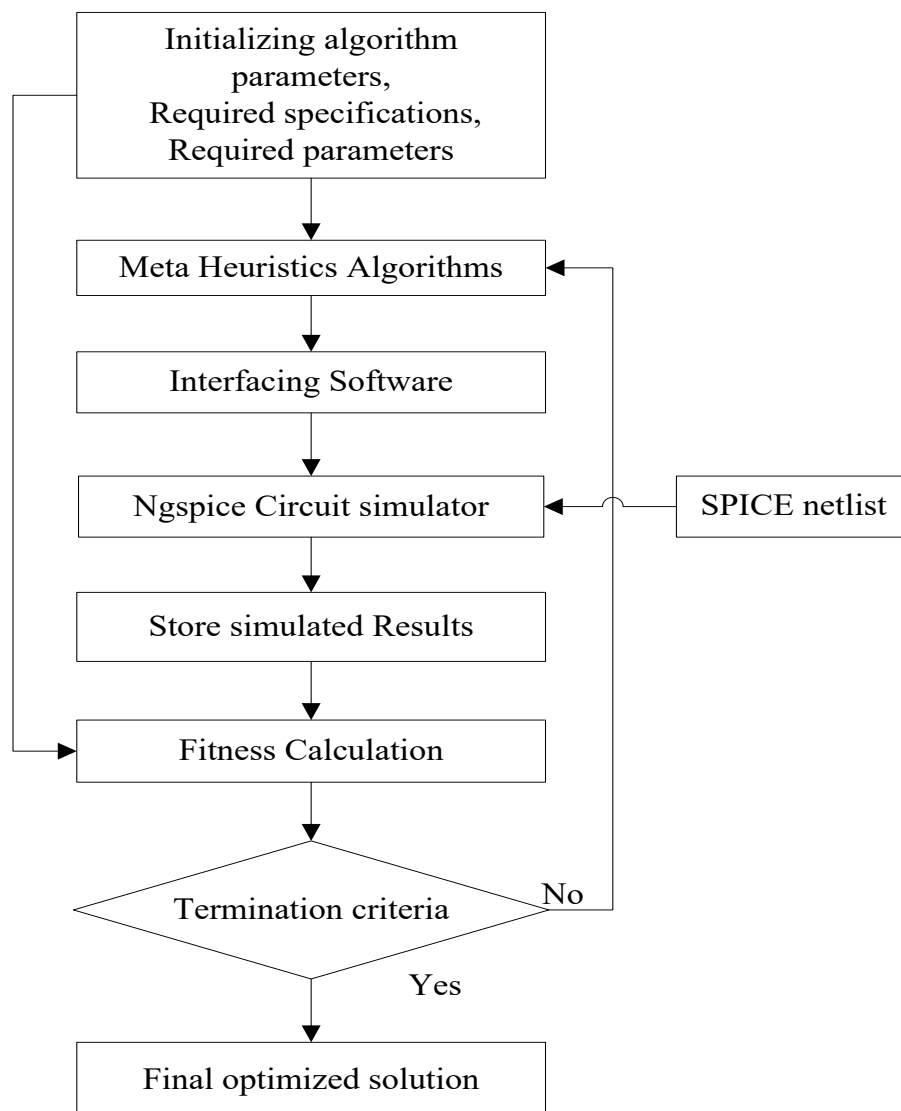
| Sr. No. | Required Specifications                       |
|---------|-----------------------------------------------|
| 1       | AC voltage gain $AV > 80$ dB                  |
| 2       | Phase margin $> 60^\circ$                     |
| 3       | Unit gain bandwidth (UGB) $> 100$ MHz         |
| 4       | Power Supply Rejection Ratio (PSRR) $> 75$ dB |
| 5       | Common Mode Rejection Ration (CMRR) $> 80$ dB |
| 6       | Rise Slew Rate (RSR) $> 40$ V/us              |
| 7       | Fall Slew Rate (FSR) $> 40$ V/us              |

The performance of this optimizer depends on several factors, including the settings of the algorithm, the number of design variables, the range of those variables, and the number of specifications that need to be optimized. Table 4, presents the desired specifications such as gain, phase margin, and slew rate, which serve as optimization targets for the two-stage operational amplifier. These specifications are provided as input to the automated design framework to define the optimization objectives. Table 5 and Table 6 show the optimized parameters of the

two-stage op-amp which displays the transistor sizing and current values obtained using the ABC and PSO algorithm for the optimal op-amp design in 130 nm CMOS technology respectively. The results indicate the transistor dimensions and component values that yield the best performance. The obtained design parameters were used to meet the specifications shown in Table 7, which compares the performance of the two-stage op-amp designs optimized by ABC and PSO with the target values and compared with the previously reported work in (6,13).

**Table 5:** Optimized Parameters of Two Stage Op-Amp for the ABC Algorithm

| Design Variable                  | Variable Range                                                  | Results for 130 nm (13) | Obtained Parameters for 130 nm |
|----------------------------------|-----------------------------------------------------------------|-------------------------|--------------------------------|
| W1 / L1                          | W: 0.5 to 10 ( $\mu\text{m}$ )<br>L: 0.2 to 1 ( $\mu\text{m}$ ) | 5.4/0.6                 | 1.88/0.42                      |
| W2 / L2                          |                                                                 | 0.5/0.3                 | 0.54/0.42                      |
| W3 / L2                          |                                                                 | 4.9/0.3                 | 5.7/0.42                       |
| W4 / L3                          |                                                                 | 9.5/1.0                 | 10/0.76                        |
| W5 / L4                          |                                                                 | 2.7/0.3                 | 10/0.2                         |
| W6 / L3                          | Transistor dimensions are in $\mu\text{m}$ .                    | 2.0/1.0                 | 4.13/0.76                      |
| W7 / L5                          |                                                                 | 4.5/0.7                 | 0.54/0.2                       |
| W8 / L5                          |                                                                 | 5.0/0.7                 | 10/0.2                         |
| W9 / L3                          | 1 to 10 $\mu\text{A}$                                           | 3.9/1.0                 | 1.79/0.76                      |
| I <sub>o</sub> ( $\mu\text{A}$ ) |                                                                 | 2.8                     | 5.10                           |
| C (pF)                           |                                                                 | 0.043                   | 0.1                            |

**Figure 8:** The Automated Analog Circuit Design Environment

**Table 6:** Optimized Parameters of Two Stage Op-Amp for the PSO Algorithm

| Design Variable                  | Variable Range                                                   | Results for 130 nm (6) | Obtained Parameters for 130 nm |
|----------------------------------|------------------------------------------------------------------|------------------------|--------------------------------|
| W1 / L1                          | W: 0.5 to 10 ( $\mu\text{m}$ )<br>L: 0.13 to 1 ( $\mu\text{m}$ ) | 2.5/0.75               | 1.25/0.35                      |
| W2 / L2                          |                                                                  | 1.5/0.5                | 2.28/0.38                      |
| W3 / L2                          |                                                                  | 3.8/0.5                | 6.38/0.38                      |
| W4 / L3                          |                                                                  | 7.0/0.75               | 9.94/1                         |
| W5 / L4                          |                                                                  | 1.5/0.25               | 2.87/0.5                       |
| W6 / L3                          | Transistor dimensions are in $\mu\text{m}$ .                     | 3.0/0.75               | 2.5/1                          |
| W7 / L5                          |                                                                  | 4.0/0.75               | 7.85/1                         |
| W8 / L5                          |                                                                  | 4.0/0.75               | 5/1                            |
| W9 / L3                          | 0.01 to 10 $\mu\text{A}$                                         | 5.5/0.75               | 4.15/1                         |
| I <sub>o</sub> ( $\mu\text{A}$ ) |                                                                  | 4.5                    | 5.52                           |
| C (pF)                           |                                                                  | 0.09                   | 0.06                           |

## Results and Discussion

The two stage CMOS based operational amplifier designed using TSMC's 130nm CMOS process with required specifications as shown in Table 4. The simulation is carried out on a Core i5 1.80 GHz processor, 8GB RAM, 64 bit Ubuntu operating system. The ABC and PSO algorithms are implemented and compiled in Python, and all circuit simulations were carried out using Ngspice-26. The design variables obtained from Tables 3 and 4 were used to evaluate the final specifications, and the results were compared with previously reported work (6, 13). Table 7 shows that both the ABC and PSO algorithms meet the all target specification for the two-stage op-amp. The required voltage gain is set to above 80 dB, with ABC at 80.01 dB and PSO

slightly higher at 80.96 dB. The phase margin is better with PSO (65.97°) compared to ABC (62.44°), both above the 60° target. Unity gain bandwidth is also higher than the target, with ABC reaching 123 MHz and PSO 121 MHz. PSRR and CMRR values improved compared to earlier work, as ABC has a PSRR of 84.97 dB and CMRR of 80.60 dB, while PSO shows 75.77 dB PSRR and 83.74 dB CMRR. Both designs exceed the rise slew rate target of 40 V/ $\mu\text{s}$ , with ABC at 43.82 V/ $\mu\text{s}$  and PSO at 44.03 V/ $\mu\text{s}$ . The fall slew rate is just around the target, with ABC slightly above at 40.01 V/ $\mu\text{s}$  and PSO a little below at 39.73 V/ $\mu\text{s}$ . Both ABC and PSO algorithms successfully met all the required performance targets listed and the results were benchmarked against previous studies to validate the effectiveness of the proposed optimization framework.

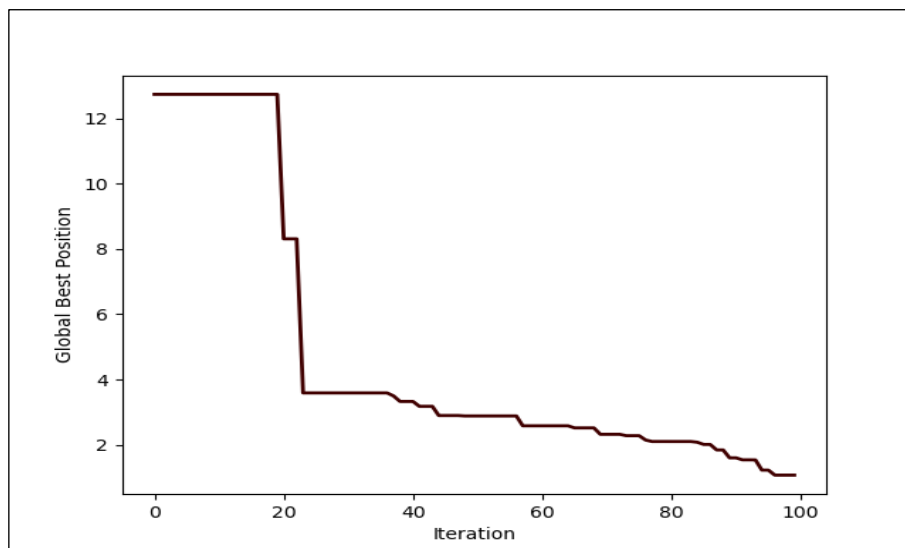
**Table 7:** Obtained Specification of Two Stage Op-Amp

| Sr. No. | Required Specifications                           | Results for 130 nm (6,13) | Results for ABC Algorithm 130 nm | Results for PSO Algorithm 130 nm |
|---------|---------------------------------------------------|---------------------------|----------------------------------|----------------------------------|
| 1       | $A_v > 80 \text{ dB}$                             | 86.16 dB                  | 80.01 dB                         | 80.96 dB                         |
| 2       | Phase Margin $> 60^\circ$                         | 61.79°                    | 62.44°                           | <b>65.97°</b>                    |
| 3       | UGB $> 100 \text{ MHz}$                           | 101 MHz                   | <b>123 MHz</b>                   | 121 MHz                          |
| 4       | PSRR $> 75 \text{ dB}$                            | ---                       | <b>84.97 dB</b>                  | 75.77 dB                         |
| 5       | CMRR $> 80 \text{ dB}$                            | ---                       | 80.60 dB                         | <b>83.74 dB</b>                  |
| 6       | Rise Slew rate (RSR) $> 40 \text{ V}/\mu\text{s}$ | 50.33 V/ $\mu\text{s}$    | 43.82 V/ $\mu\text{s}$           | 44.03 V/ $\mu\text{s}$           |
| 7       | Fall Slew rate (FSR) $> 40 \text{ V}/\mu\text{s}$ | 37.79 V/ $\mu\text{s}$    | 40.01 V/ $\mu\text{s}$           | 39.73 V/ $\mu\text{s}$           |

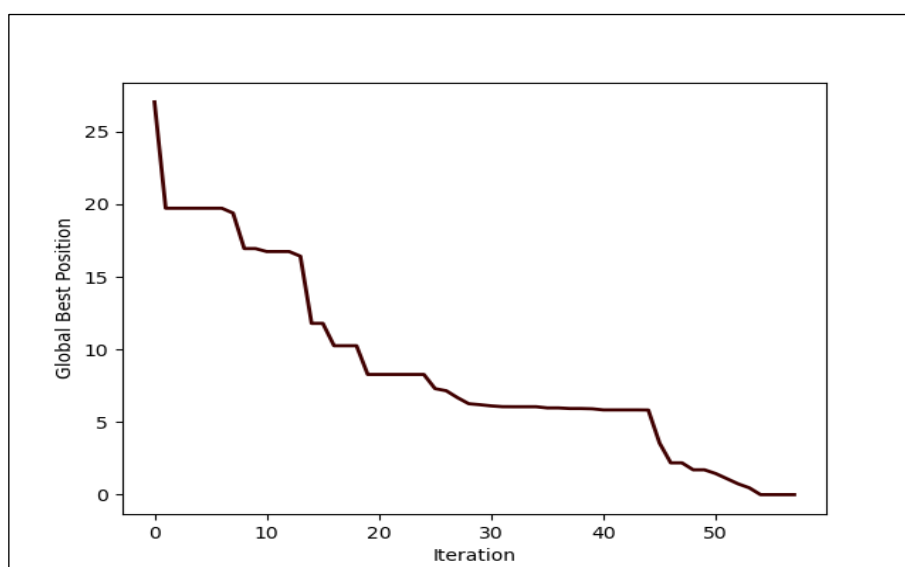
In addition to convergence speed, a comparative evaluation was carried out with respect to scalability, accuracy, and efficiency. The results show that PSO generally achieves faster convergence and higher accuracy in reaching the target specifications, making it more efficient when computational time is a critical factor. However, ABC demonstrates greater robustness and scalability, as it maintains stable performance even when the number of design variables or specifications increases. The main trade-off, therefore, lies between speed PSO and robustness ABC. Figure 9 displays how the ABC algorithm minimizes the fitness function across iterations for op-amp design and Figure 10, shows PSO's faster convergence compared to ABC. This balanced view highlights

both the strengths and limitations of the two approaches, offering guidance for their use in practical analog circuit optimization depending on design priorities.

The present work has focused on optimizing a two-stage CMOS operational amplifier. The future scope of this research is to demonstrate scalability from relatively simple circuits, like the two-stage OTA, to more complex analog blocks and system-level designs. This would provide deeper insights into the robustness of the proposed optimization approach and further establish its value for practical analog and mixed-signal circuit design, particularly in the context of low-power and battery-operated applications.



**Figure 9:** Convergence Graph of Op-Amp for the ABC Algorithm



**Figure 10:** Convergence Graph of Op-Amp for the PSO Algorithm

## Conclusion

A set of widely used unimodal and multimodal benchmark functions were employed to evaluate the performance of optimization algorithms, with details on their mathematical formulations, search spaces, and global optima. Leveraging this evaluation, an automated optimization approach based on metaheuristic Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) algorithms was applied to optimize a CMOS two-stage operational amplifier designed in TSMC's 130nm technology node. Both ABC and PSO algorithms effectively optimized the two-stage op-amp achieving overall performance metrics. Achieving key performance metrics including a voltage gain of 80.96 dB, phase margin of 65.97°, unity gain bandwidth of 121 MHz, PSRR of 75.77 dB, CMRR of 83.74 dB, and rise slew rate of 44.03 V/ $\mu$ s, surpassing all target specifications except fall slew rate of 39.73 V/ $\mu$ s using PSO algorithm. The simulation results for these metrics were compared against the target specifications and summarized. These findings demonstrate the effectiveness and robustness of the proposed metaheuristic optimization framework for analog circuit design which satisfy all the specified performance criteria.

## Abbreviations

ABC: Artificial Bee Colony,  $A_v$ : Voltage Gain, CMRR: Common Mode Rejection Ratio, GA: Genetic Algorithms, Op-amp: Operational Amplifier, PSO: Particle Swarm Optimization, PSRR: Power Supply Rejection Ratio, UGB: Unity Gain Bandwidth.

## Acknowledgement

Authors are thankful to Gujarat Technological University and L.D. College of Engineering, Ahmedabad for research facilities.

## Author Contributions

Sureshbhai L Bharvad: methodology, Python code development, preparation of the automated design framework integrating the algorithms with the NGSpice simulator, writing of the original draft, Pankajkumar P Prajapati: supervision, oversaw the investigation, contributed to writing, reviewing, and editing of the manuscript.

## Conflict of Interest

The authors declare that there are no conflicts of interest or competing interests related to this work.

## Declaration of Artificial Intelligence

### (AI) Assistance

The authors declare no use of Artificial intelligence (AI) for the write-up of the manuscript.

### Ethics Approval

This research does not involve human or animal participants; therefore, ethics approval not required.

### Funding

No external funding was used during this research.

## References

1. Thakker RA, Sathe C, Sachid AB, *et al.* A Novel Table-Based Approach for Design of FinFET Circuits. *IEEE Trans Comput-Aided Des Integr Circuits Syst.* 2009 July;28(7):1061–70.
2. Koh HY, Sequin CH, Gray PR. OPASYN: a compiler for CMOS operational amplifiers. *IEEE Trans Comput-Aided Des Integr Circuits Syst.* 1990;9(2):113–25.
3. Massoud Y, Nieuwoudt A, Ragheb T. Expression of Concern: Automated Design Solutions for Fully Integrated Narrow-Band Low Noise Amplifiers. In: 2006 6th International Workshop on System on Chip for Real Time Applications, Cairo, Egypt. 2006:109–14. <https://ieeexplore.ieee.org/abstract/document/4155271/>
4. Savio A, Colalongo L, Quarantelli M, Kovacs-Vajna ZM. Automatic Scaling Procedures for Analog Design Reuse. *IEEE Trans Circuits Syst Regul Pap.* 2006 Dec;53(12):2539–47.
5. Sapatnekar SS, Rao VB, Vaidya PM, *et al.* An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization. *IEEE Trans Comput-Aided Des Integr Circuits Syst.* 1993 Nov;12(11):1621–34.
6. Thakker RA, Baghini MS, Patil MB. Low-Power Low-Voltage Analog Circuit Design Using Hierarchical Particle Swarm Optimization. In: 2009 22nd International Conference on VLSI Design. New Delhi, India: IEEE; 2009:427–32. <https://ieeexplore.ieee.org/abstract/document/4749710/>
7. Barros M, Guilherme J, Horta N. Analog Circuits Optimization Based on Evolutionary Computation Techniques. *Integration.* 2010 Jan;43(1):136–55.
8. Vural RA, Yildirim T. Analog Circuit Sizing via Swarm Intelligence. *AEU - Int J Electron Commun.* 2012 Sept;66(9):732–40.
9. Zhu G, Kwong S. Gbest-guided Artificial Bee Colony Algorithm for Numerical Function Optimization. *Appl Math Comput.* 2010 Dec;217(7):3166–73.
10. Pasandi G, Pratty S, Forsyth J. AISYN: AI-driven Reinforcement Learning-Based Logic Synthesis Framework. *arXiv.* 2023. <https://arxiv.org/abs/2302.06415>
11. Yin Y, Wang Y, Xu B, Li P. ADO-LLM: Analog Design Bayesian Optimization with In-Context Learning of Large Language Models. 2024:1–9. <https://arxiv.org/abs/2406.18770>
12. Karaboga D, Basturk B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *J Glob Optim.*

- 2007 Oct 9;39(3):459–71.
13. Patel SJ, Thakkar RA. An Efficient Artificial Bee Colony Algorithm and Analog Circuit Design Environment. WSEAS Trans CIRCUITS Syst. 2017;16(1):108–22.
  14. Kennedy J, Eberhart R. Particle Swarm Optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. Perth, WA, Australia: IEEE. 1995;4:1942–8.
  15. Butani RC, Gajjar BD, Thakker RA. Performance Evaluation of Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) Algorithm. 2011;108–12. <https://www.academia.edu/download/11490593/232.pdf>
  16. Shi Y, Eberhart RC. Empirical Study of Particle Swarm Optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99. Washington, DC, USA: IEEE. 1999;3:1945–50.
  17. Jamil M, Yang XS. A literature Survey of Benchmark Functions for Global Optimization Problems. Int J Math Model Numer Optim. 2013;4(2):150.
  18. Shah H, Tairan N, Garg H, Ghazali R. Global Gbest Guided-Artificial Bee Colony Algorithm for Numerical Function Optimization. Computers. 2018 Dec 7;7(4):69.
  19. Trelea IC. The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett. 2003;85(6):317–325.
  20. Allen P, Holberg D. CMOS Analog Circuit Design. 2nd ed. New York: Oxford University Press. 2013. <https://books.google.com/books?hl=en&lr=&id=N42wM6vtAhkC&oi=fnd&pg=PP2&dq=Allen+P,+Holberg+D.+CMOS+Analog+Circuit+Design.+2nd+ed.+New+York:+Oxford+University+Press.+2013.+&ots=RwqiD2885S&sig=w2x5M4qq3Qcr5L0RzhG1SjslW4Y>
  21. Priyanka Kakoty. Design of a High Frequency Low Voltage CMOS Operational Amplifier. Int J VLSI Des Commun Syst. 2011 Mar 24;2(1):73–85.
  22. Nageshwarrao D, Kumar KS, Rao YR, Jyothi G. Implementation and Simulation of CMOS Two Stage Operational Amplifier. Int J Adv Eng Technol. 2013 Jan;5(2):162–7.
  23. Singh CL, Anandini C, Gogoi AJ. Automated sizing of low-noise CMOS analog amplifier using ALCPSO optimization algorithm. J Inf Optim Sci. 2017;39(1):99–111.
  24. Budak A, Gandara M, Shi W, Pan D-Z, Sun N, Liu B. An efficient analog circuit sizing method based on machine learning assisted global optimization. IEEE Trans Comput Aided Des Integr Circuits Syst. 2021;41(5):1209–1221.
  25. Shreeharsha KG, Siddharth RK, Vasantha MH, Nithin Kumar YB. An error bound particle swarm optimization for analog circuit sizing. IEEE Access. 2024;12:50126–50136.
  26. Patel SJ, Thakker RA. Parasitic-Aware Automatic Analog CMOS Circuit Design Environment. In: 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID). Delhi, NCR, India: IEEE. 2019:245–50. <https://www.computer.org/csdl/proceedings-article/vlsid/2019/040900a245/1a3wXgL3KE0>
  27. Shahraki NS, Mohammadi A, Mohammadi-Esfahrood S, Zahiri SH. Improving the performance of analog integrated circuits using multi-objective metaheuristic algorithms. In: Proceedings of the 5th Conference on Knowledge Based Engineering and Innovation (KBEI); Tehran, Iran. IEEE; 2019. p.822–826.

**How to cite:** Bharvad SL, Prajapati PP. Automated analog circuit design by evolutionary algorithms. Int Res J Multidiscip Scope. 2025; 6(4):1075-1088. doi: 10.47857/irjms.2025.v06i04.06605