IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 7 February 2024, accepted 25 March 2024, date of publication 5 April 2024, date of current version 12 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3385491

RESEARCH ARTICLE

An Error Bound Particle Swarm Optimization for Analog Circuit Sizing

K. G. SHREEHARSHA[1], R. K. SIDDHARTH[2], (Member, IEEE), M. H. VASANTHA[3], (Member, IEEE), AND Y. B. NITHIN KUMAR[3], (Senior Member, IEEE)

[1]Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
[2]Department of Mechatronics Engineering, Parul Institute of Technology, Parul University, Vadodara 391760, India
[3]Department of Electronics and Communication Engineering, National Institute of Technology Goa, Goa 403401, India

Corresponding author: K. G. Shreeharsha (shreeharsha.kg@manipal.edu)

This work was supported by the Research and Development Project through the Department of Science and Technology-Science and Engineering Research Board (DST-SERB), Scheme of Government of India, under Project CRG12019/00l761.

ABSTRACT An Error-Bound Particle Swarm Optimization (EB-PSO) is proposed in this work. The objective function is evaluated for each particle in each iteration. The velocity update equation is modified by introducing two new parameters ζ_1 and ζ_2. These parameters varies exponentially, within the bounds $(\zeta_{1,min}, \zeta_{2,min})$ and $(\zeta_{1,max}, \zeta_{2,max})$, with respect to the number of iterations. Initially, a higher value of ζ_2 and minimum value of ζ_1 is chosen to facilitate a global search. Once the global error (ε_2) is less than the desired value, ζ_1 is allowed to increase from its minimum value and ζ_2 is held constant at $\zeta_{2,max}$. This leads to local exploitation of the search space. The proposed algorithm is implemented on Python platform. To verify the effectiveness of the proposed EB-PSO algorithm in analog circuit sizing, a case study on the performance and optimization of two-stage op-amp is presented, whose validation is done in Cadence-Virtuoso environment at 45-nm CMOS technology. The results show that the proposed EB-PSO algorithm converges in 11 iterations for two-stage op-amp, whereas it takes 23, 29, and 41 iterations to converge for conventional GA, DE, and PSO algorithms respectively.

INDEX TERMS Analog circuit sizing, particle swarm optimization (PSO), constrained optimization.

I. INTRODUCTION

Today, real-world domains such as industrial electronics and telecommunications face many complex challenges. These problems are difficult to solve due to constraints, non-convexity, and uncertainties. Formulating an optimization problem that incorporates constraints, objectives, design variables, and iteration count is a difficult task. However, over the last few decades, various optimization techniques have been widely used to solve these complex real-world problems. In general, optimization methods can be broadly classified into two categories: deterministic and stochastic [1]. Deterministic method involves differentiation based approach in the solving process. In this approach, accuracy in achieving the optimum solution is guaranteed. However, this method requires a huge amount of computation time, which makes them impractical in solving complex problems. On the contrary, stochastic methods involve randomness in one or more variables while maximizing or minimizing an objective function in the solving process. Today, stochastic methods are popularly used as optimizer, due to their potential and adaptability in incorporating the domain specific knowledge. Even though the optimum solution is not guaranteed in this approach, superior solutions with desired accuracy are achieved in a reasonable amount of time, making it a suitable choice for the designer. Thus, the computational time is one of the most pertinent advantage in this approach over deterministic methods.

Various heuristic [2], [3], [4], [5], [6], [7], [8], [9] and meta-heuristic algorithms [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] involve stochastic based approach in finding the optimum solutions for high

The associate editor coordinating the review of this manuscript and approving it for publication was Jagadheswaran Rajendran.

computational problems. Heuristic approach is specific to a particular optimization problem whereas meta-heuristic is a generic approach, which can be adapted in solving diverse optimization problems. The drawback of the heuristics to be problem-dependent, limits its usage in diverse field of applications [10]. Meta-heuristic algorithms include various optimization techniques like, among many others, particle swarm optimization (PSO) [22], [23], [24], [25], [26], [27], genetic algorithms [28], [29] and differential evolution [30], [31]. Nature based metaheuristic algorithms are gaining high recognition in solving computationally expensive optimization problems [21]. PSO is a derivative-free population-based technique, which is used to solve discrete as well as continuous optimization problems. It is found that it can easily fall into local optimum in a high-dimensional search space resulting in premature convergence [23]. Furthermore, it exhibits a lower convergence rate in the iterative process. To address this, a modified variant of PSO known as Historical Memory Particle Swarm Optimization (HMPSO) was employed to seek an improved solution compared to the conventional PSO method [32]. PSO using fuzzy algorithm was used to overcome premature convergence, which is typically found in conventional PSO [33]. A hybrid PSO was presented in [34], in which the convergence rate was improved by linearly decreasing the inertia weight.

A summary of recently developed algorithms addressing optimization problems includes the Chimp Optimization Algorithm (ChoA), inspired by the cooperative intelligence of chimpanzees during group hunts [35]. Notably, ChoA is characterized by the absence of initial assumptions and population dependency, prompting the development of the Improved Chimp-Spotted Hyena Optimizer (ICHIMP-SHO). To enhance the optimization capabilities of the original Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA), two new algorithms were introduced: the Gravitational Particle Swarm Optimization Algorithm (GPSOA) [36] and Sequential Hybrid PSO-GSA (SHPSO) [37]. These algorithms aim to leverage the exploration strength of PSO and the exploitation advantage of GSA to overcome their respective limitations. Additionally, the Sparrow Search Algorithm (SSA) was proposed, emphasizing robust search capabilities and rapid convergence speed [38]. SSA is designed to efficiently explore and identify optimal parameters in optimization tasks.

This paper introduces a modified approach to PSO called Error-Bound PSO (EB-PSO), which involves the inclusion of two parameters, ζ_1 and ζ_2, into the conventional PSO. These parameters vary exponentially with iterations count, and are constrained within the range of $(\zeta_{1,min}, \zeta_{2,min})$ and $(\zeta_{1,max}, \zeta_{2,max})$. Additionally, their variation is determined by the local and global errors, which are calculated at each iteration. This dependency on the errors results in an improved convergence rate. To validate the efficacy of the EB-PSO algorithm, the paper includes a case study on two-stage op-amp.

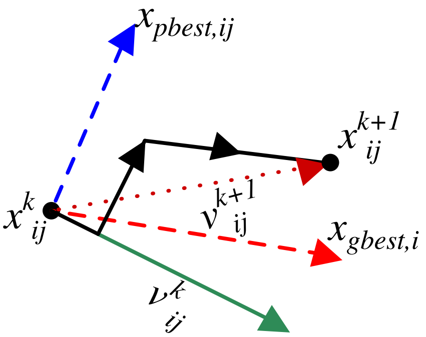
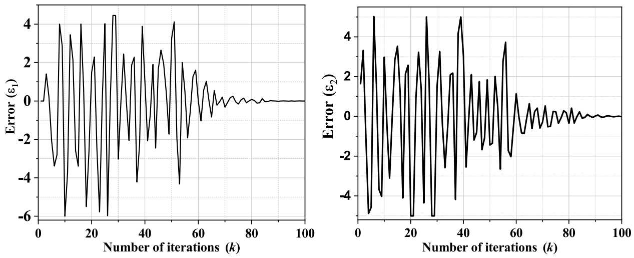


FIGURE 1. Operation of particle swarm optimization.



(a)

(b)

FIGURE 2. Variation of (a) local error (ε_1) and (b) global error (ε_2) with iterations count.

The paper is organized as follows. Section II explores the conventional PSO algorithm. The proposed EB-PSO algorithm is discussed in Section III. Section IV presents the simulation results and includes three case studies. The paper is concluded in Section V.

II. BACKGROUND THEORY

A good exploitation ability ensures that the algorithm converges faster to a global optimum solution and good exploration results in no premature convergence. Considering a n-dimensional search space, the particles are initialized with uniform random velocity and position in the search space. Let for a particle j, the position and velocity be represented by x_j and v_j respectively as shown in Fig. 1, where $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ and $v_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$. Let x_{pbest} be the personal best position for this particle j and x_{gbest} be the global best position of the group. The velocity of the particle is calculated using the following equation:

$$v_{ij}^{k+1} = w * v_{ij}^{k} + c_1 * r_1(x_{pbest,ij}^{k} - x_{ij}^{k}) + c_2 * r_2(x_{gbest,i}^{k} - x_{ij}^{k}) \tag{1}$$

the variable ‘k’ denotes the iteration count, ‘i’ ranges from 1 to n, and ‘j’ ranges from 1 to p. The values ‘p’ and ‘n’ represent the total number of particles and variables, respectively. The coefficient of cognitive acceleration, denoted as ‘c_1’, aids in exploring the search space, while the coefficient of social acceleration, ‘c_2’, facilitates exploiting the search space. Additionally, ‘r_1’ and ‘r_2’ are random uniform values generated in the range of 0 to 1. Each particle’s position is

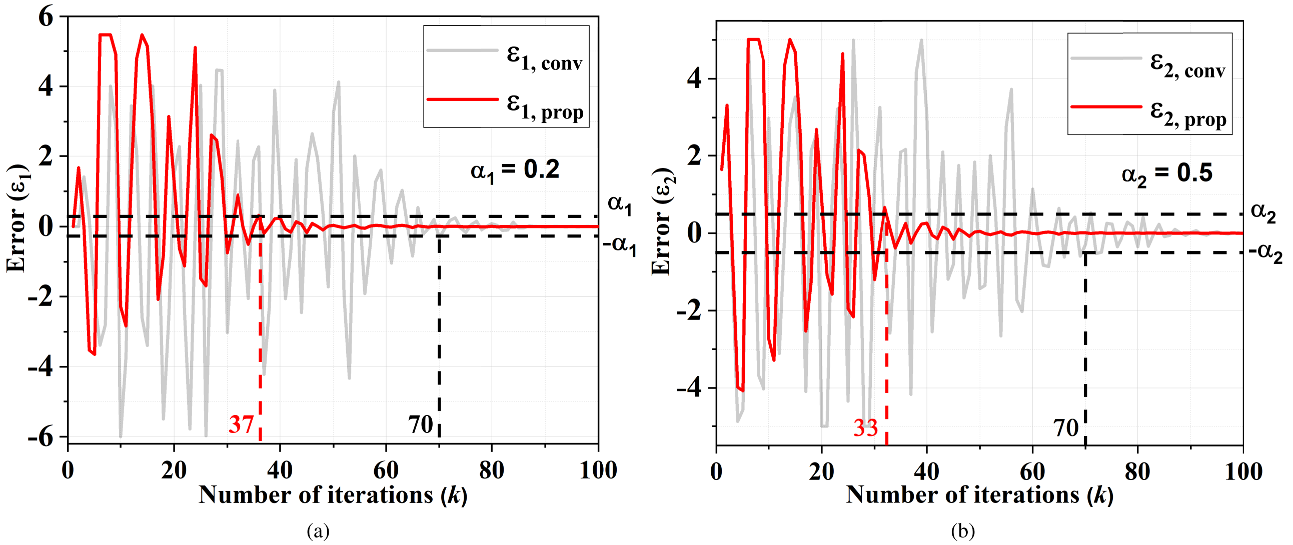


FIGURE 3. Variation of local error (ε_1) and global error (ε_2) with iterations count for the objective function $f_1(x)$.

subsequently updated using equation (2).

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{2}$$

To enhance the PSO algorithm's search capability, an inertia weight 'w' is introduced in the velocity equation (refer eq. (1)). The value of 'w' determines the impact of the previous velocity and, in turn, affects the particle's speed in converging. Specifically, w = w*wdamp, where a larger value of 'w' facilitates global search, while a smaller value enables local search. The search terminates when it reaches the maximum number of iterations or meets the desired error criteria. This work explores improving the convergence rate by examining the error between the local best ($x_{pbest,ij}$) and global best (($x_{gbest,i}$)) with respect to the particle's current position (x_{ij}). Thus, the error between $x_{pbest,ij}$ and x_{ij} is termed as the local error (ε_1), i.e.

$$\varepsilon_1 = x_{pbest,ij} - x_{ij} \tag{3}$$

Similarly, the error between $x_{gbest,i}$ and x_{ij} is termed as the global error (ε_2), i.e.

$$\varepsilon_2 = x_{gbest,i} - x_{ij} \tag{4}$$

Here, the local error refers to the error of a particle's current position in its local neighborhood, which helps the particle to update its position and velocity by considering the best position found by its neighbors in the current iteration. On the other hand, the global error is the error of the best solution found so far by all particles, which acts as a guide for the particles to converge towards optimal solution. To understand the behaviour of the local and global errors, consider an example of one-dimensional objective function as shown in eq. (5).

$$\begin{aligned} &\text{Minimize} \quad f_1(x) = x^2 \\ &s.t. \quad -5 \le x \le 5 \end{aligned} \tag{5}$$

The objective function in (5) is minimized using a conventional PSO algorithm and local as well as global errors are obtained with iterations count (k), which are plotted in Fig. 2. The errors vary and eventually stabilize after several iterations, signifying the convergence of the optimization algorithm. The function's value obtained after the convergence is the minimum value achieved using the optimization algorithm. It can be observed that both the errors take 98 iterations to settle. This necessitates a more number of iterations to optimize an objective function.

III. ERROR-BOUND PARTICLE SWARM OPTIMIZATION

In traditional PSO, the update of particle velocity follows the equation expressed in equation (1). It was noted that the velocity is influenced by the value c_1, c_2, r_1 and r_2. The global and local errors exhibit fluctuations with every iteration. During the initial search phase, the positions of the particle may deviate significantly from their global best, resulting in a large error. As the particles exchange information about their best possible location, the position error gradually decreases, but conventional PSO algorithms tend to have a slow convergence process. To expedite convergence, prior research has recommended the use of a inertia weight with time-decreasing behaviour. Nevertheless, this method still necessitates a considerable number of iterations to achieve convergence [39].

Concerning the design of analog circuits, there is a notable increase in both complexity and design cycle time. Consequently, the proposed EB-PSO algorithm aims to alleviate the design cycle time. This study delves into the time complexity of the EB-PSO algorithm, specifically focusing on a variant that introduces the parameters ζ_1 and ζ_2 to influence the search process. The manner in which ζ_1 varies plays a crucial role in shaping the behavior of the EB-PSO algorithm. Opting for linear variations of ζ_1 provides

Algorithm 1 Proposed EB-PSO Algorithm: A Step-by-Step Pseudo-Code

```
begin
Initialization: Number of Dimensions (n); Acceleration Coefficients ($c_1$ and $c_2$); Population Size (p); Maximum Number of Iterations (N); inertia weight (w); $\alpha_{min}$; $\alpha_{max}$; $\alpha_1$; $\alpha_2$ establishes the boundaries for the design parameters.
for all particle do
    Generate $x_i$ randomly and assess its fitness, denoted as $f(x_i)$.
    if current fitness value is superior to its previous personal best (pbest) then
        Set $pbest_i = x_i$ and $f(pbest_i) = f(x_i)$
for all particle do
    if $f(pbest_i) < f(gbest)$ then
        $f(gbest) = f(pbest_i)$
while maximum iterations are not reached do
    Evaluate $\varepsilon_1$ and $\varepsilon_2$ using eq. (3) and (4)
    if $|\varepsilon_2| > \alpha_2$ then
        $\zeta_2 = \zeta_{2,max} * e^{-\frac{k}{N}}$
        $\zeta_1 = \zeta_{1,min}$
    else if $|\varepsilon_1| > \alpha_1$ then
        $\zeta_1 = \zeta_{1,max} - (\zeta_{1,max} - \zeta_{1,min})e^{-\frac{k}{N}}$
        $\zeta_2 = \zeta_{2,min}$
    else
        $\zeta_1 = \zeta_{1,max}$
        $\zeta_2 = \zeta_{2,min}$
    Evaluate the velocity of the particle using eq. (6)
    Evaluate its position using eq. (2)
    if pbest meets all the specified constraints and bounds then
        $pbest_i = x_i(k+1)$
        $f(pbest) = f(x_i(k+1))$
    if $f(pbest) < f(gbest)$ then
        $gbest = pbest_i$
        $f(gbest) = f(pbest_i)$
    k = k + 1
end
return gbest
```

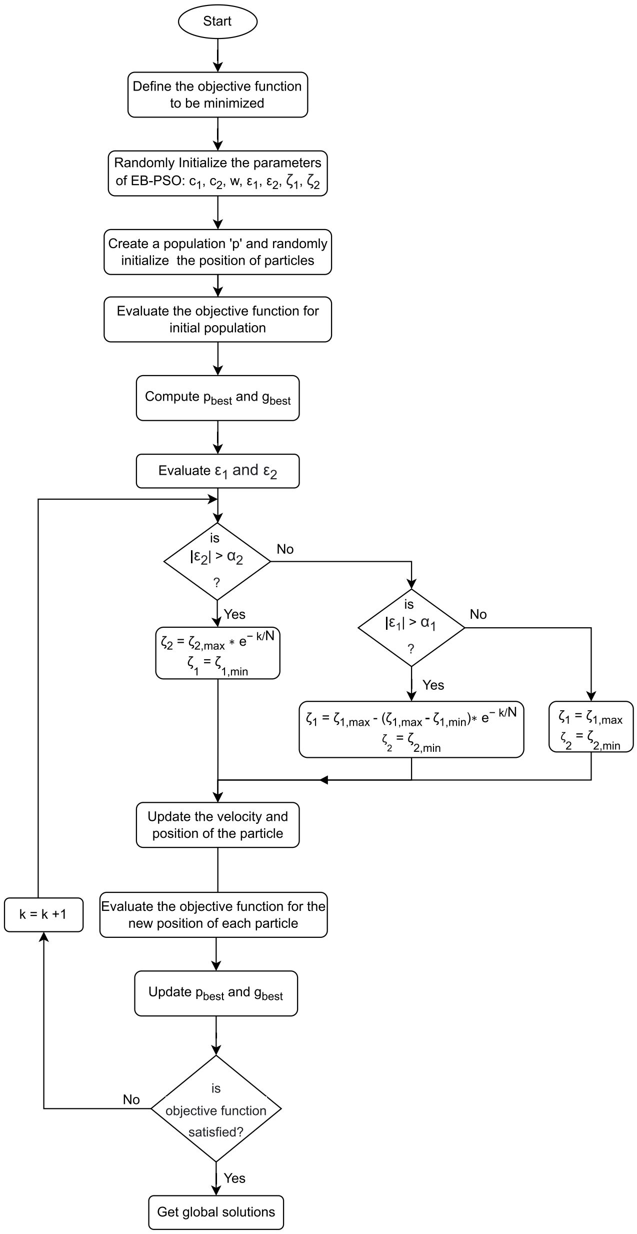

Start
Define the objective function to be minimized
Randomly Initialize the parameters of EB-PSO: c_1, c_2, w, ε_1, ε_2, ζ_1, ζ_2
Create a population 'p' and randomly initialize the position of particles
Evaluate the objective function for initial population
Compute p_{best} and g_{best}
Evaluate ε_1 and ε_2
is $|\varepsilon_2| > \alpha_2$?
No
Yes
$\zeta_2 = \zeta_{2,max} * e^{-k/N}$
$\zeta_1 = \zeta_{1,min}$
is $|\varepsilon_1| > \alpha_1$?
No
Yes
$\zeta_1 = \zeta_{1,max} - (\zeta_{1,max} - \zeta_{1,min}) * e^{-k/N}$
$\zeta_2 = \zeta_{2,min}$
$\zeta_1 = \zeta_{1,max}$
$\zeta_2 = \zeta_{2,min}$
Update the velocity and position of the particle
k = k +1
Evaluate the objective function for the new position of each particle
Update p_{best} and g_{best}
is objective function satisfied?
No
Yes
Get global solutions

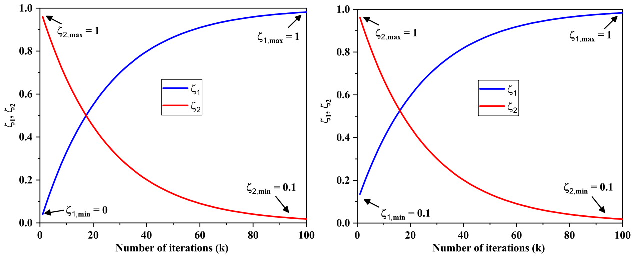
FIGURE 4. Flowchart of EB-PSO algorithm.

enhanced control over the transition between exploration and exploitation, offering a gradual shift from global exploration to local exploitation. However, the algorithm's behavior is significantly affected by the slope and intercept of the linear function. While such gradual variations prove beneficial for simpler problems, exponential variations of ζ_1 are more conducive to rapid convergence in the initial stages of the algorithm for complex and high-dimensional issues like analog circuit design. Nevertheless, an excessively high decreasing rate may result in premature convergence and entrapment in local optima if exploration diminishes too swiftly. To address this, the present study introduces the utilization of error-dependent coefficients for global and local errors to enhance convergence speed. The parameters ζ_1 and ζ_2 vary based on local and global errors. These coefficients, denoted as ζ_1 and ζ_2, respectively, enable an increased particle velocity directed towards minimizing the error, thereby enhancing the learning rate as expressed in eq. (6).

$$\begin{aligned} v_{ij}^{k+1} &= w * v_{ij}^{k} + \zeta_1 * c_1 * r_1(x_{pbestij}^{k} - x_{ij}^{k}) \\ &\quad + \zeta_2 * c_2 * r_2(x_{gbesti}^{k} - x_{ij}^{k}) \end{aligned} \tag{6}$$

where ζ_1 and ζ_2 are defined as,

$$\zeta_1 = \begin{cases} \zeta_{1,max} - (\zeta_{1,max} - \zeta_{1,min})e^{-\frac{k}{N}}; & |\varepsilon_1| > \alpha_1, |\varepsilon_2| < \alpha_2 \\ \zeta_{1,max}; & |\varepsilon_1| < \alpha_1, |\varepsilon_2| < \alpha_2 \end{cases}$$



(a) $\zeta_{1,min} = 0$

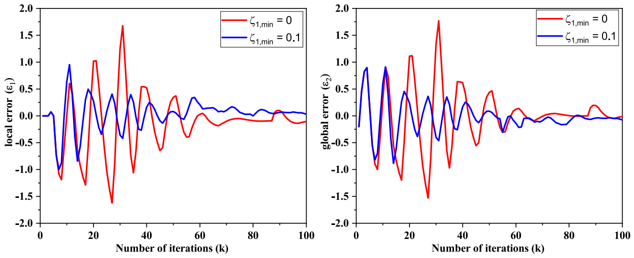
(b) $\zeta_{1,min} = 0.1$

FIGURE 5. Variation of ζ_1 and ζ_2 with iterations count (k) for (a) $\zeta_{1,min}$ = 0 and (b) $\zeta_{1,min}$ = 0.1.

$$\zeta_2 = \begin{cases} \zeta_{2,max} * e^{-\frac{k}{N}}; & |\varepsilon_2| > \alpha_2 \\ \zeta_{2,min}; & |\varepsilon_2| < \alpha_2 \end{cases} \tag{7}$$

Here, ($\zeta_{1,min}$, $\zeta_{2,min}$) and ($\zeta_{1,max}$, $\zeta_{2,max}$) denotes the lower and upper bound for the parameters ζ_1 and ζ_2 respectively. Also, α_1 and α_2 represents minimum tolerable value of error for local and global error respectively. The value of α_1 and α_2 is dependent on the type of objective function and the bounds associated with the design variables used in the objective function. During the initial search when global error is high (i.e. $|\varepsilon_2| > \alpha_2$), ζ_2 is set to $\zeta_{2,min}$ and is exponentially decreased with each iteration as seen from eq. (7). Meanwhile, ζ_1 is set to a minimum as the velocity is defined by exploration. When the global error falls below a threshold of α_2 (indicating the end of the exploration process), ζ_1 starts increasing exponentially, while ζ_2 remains fixed to a low value ($\zeta_{2,min}$). This marks the beginning of the exploitation process, which continues until the local error reaches a desired low threshold of α_1, indicating the end of optimization process. The use of error-dependent coefficients for global and local errors create a piece-wise functions for the parameters ζ_2 and ζ_1.

Algorithm 31 outlines the functioning of the proposed PSO. The evaluation of the objective function for each particle takes place in every iteration. Here, if the fitness value of the particle is less than its best value, then the local best (x_{pbest}) is updated with the particle current position. The particle having the best fitness value among all other particles is chosen to be global best (x_{gbest}). The local and global errors are then calculated. Initially, the global error (ε_2) is evaluated. If $|\varepsilon_2|$>α_2, then the ζ_2 value is chosen to be an exponentially decreasing function of iteration count (k) as shown in eq. (7) and ζ_1 is set to $\zeta_{1,min}$. Once the global error is within the desired limits, the local error is then evaluated. If $|\varepsilon_1|$>α_1, then the ζ_1 is chosen to be an exponentially increasing function of iteration count (k) as shown in eq. (7). Also, ζ_2 is set to $\zeta_{2,max}$. For the next iteration k, the entire swarm is updated in every dimension by updating the velocity of each particle and its corresponding position using eq. (6) and (2) respectively. The proposed algorithm is then used to examine the behaviour of ε_1 and ε_2 for the objective function ($f_1(x)$) defined in eq. (5) and the results are plotted in Fig. 3. It is observed that the proposed PSO algorithm takes 33 iterations for ε_2 and 37 iterations for ε_1 to settle. It is important to note that ε_1 settles after ε_2 as the global error ε_2 is evaluated before the local error ε_1. The flowchart representing the steps followed for proposed EB-PSO algorithm is depicted in Fig. 4.



(a)

(b)

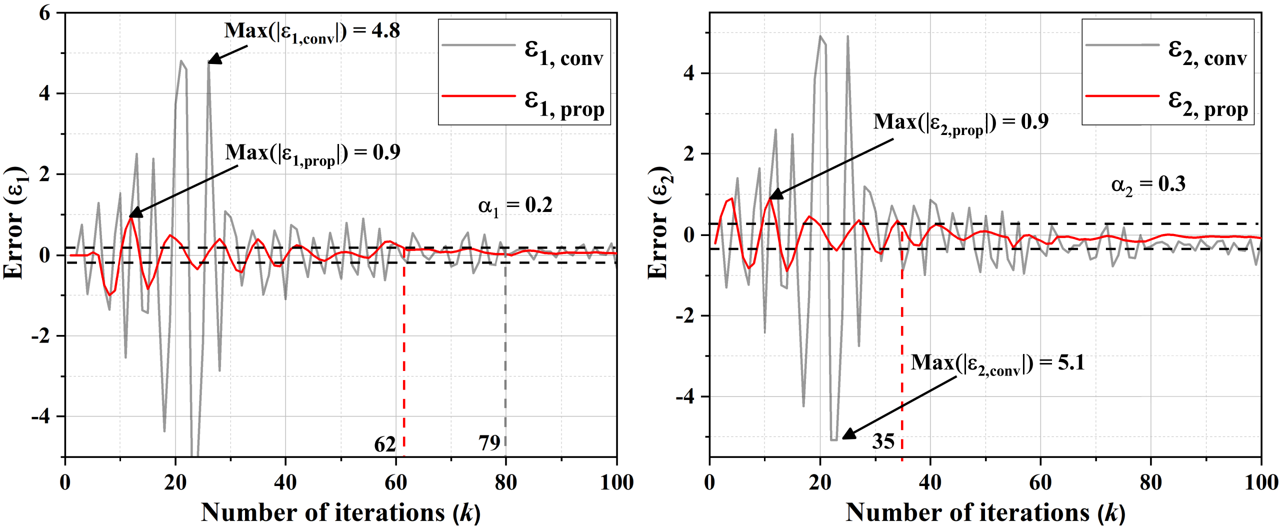
FIGURE 6. Variation of (a) ε_1 and (b) ε_2 with iterations count(k) for $\zeta_{1,min}$ = 0 and $\zeta_{1,min}$ = 0.1.

A. CHOICE OF ($\zeta_{1,MIN}$, $\zeta_{2,MIN}$) AND ($\zeta_{1,MAX}$, $\zeta_{2,MAX}$)

In the proposed EB-PSO algorithm, ($\zeta_{1,min}$, $\zeta_{2,min}$) and ($\zeta_{1,max}$, $\zeta_{2,max}$) defines the lower and upper range of ζ_1 and ζ_2 values. In Fig. 5(a), if $\zeta_{1,min}$ is set to zero, the local error is eliminated during the initial search, and the velocity update equation is only dependent on the global error. This causes the particles to move towards the global best solution in each iteration, limiting the search space exploration to the area around the global best solution. Consequently, the algorithm may not be able to discover potentially better solutions in other parts of the search space, leading to slower convergence. However, when $\zeta_{1,min}$ is set to 0.1 in Fig. 5(b), there is a minimum dependency on the local error term during the initial search, resulting in a significant improvement in convergence rate. The effectiveness of this concept is demonstrated by evaluating objective function $f_1(x)$ under both scenarios, and the results plotted in Fig. 6 show that setting $\zeta_{1,min}$ to 0.1 leads to faster convergence of local and global errors. Additionally, once the exploitation process is completed, the local error term is sufficient enough to explore near the global best solution to find optimal solution. Thus, $\zeta_{1,max}$ and $\zeta_{2,min}$ are chosen to be 1 and 0 respectively. To maintain balance between the local and global error, $\zeta_{2,max}$ is set to 1.

B. CHOICE OF α_1 AND α_2

In the proposed EB-PSO method, the values of α_1 and α_2 determine the minimum allowable error for local and global errors, respectively. α_1 specifies the desired level of accuracy for the objective function, while α_2 is crucial for determining the rate of convergence. The appropriate values of α_1 and α_2 depend on several factors, including the bounds of the objective function, the desired level of accuracy, and the specifications to be met. Ultimately, it is up to the circuit designer's intuition and experience to choose the values of α_1 and α_2 that will facilitate faster convergence. It should be noted that α_1 is kept smaller than α_2, as the global error is evaluated before the local error.



(a)

(b)

FIGURE 7. Variation of local error (ε_1) and global error (ε_2) with iterations count for the objective function $f_2(x)$.

TABLE 1. Parameters employed in the proposed EB-PSO algorithm.

Parameter	Values
(w, w_{damp})	(1 , 0.995)
Population Size (p)	35
Maximum iterations (N)	100
Dimension (n)	10 (Fig. 9)
c_1	3
c_2	1

TABLE 2. Technology parameters.

Parameter	Values
Technology	45 nm
Supply Voltage (V_{DD})	1.8 V
Kp	294 $\mu A/V^2$
Kn	334 $\mu A/V^2$
|Vthp|	0.334 V
Vthn	0.4 V
λ_n	0.2 V^{-1}
λ_p	0.1 V^{-1}

IV. RESULTS AND DISCUSSIONS

The proposed EB-PSO algorithm is executed on the Python platform utilizing TensorFlow and Keras to achieve minimum local and global errors for the considered case studies. The optimization problems are performed on an Intel i7-9700 CPU, running at 3 GHz, and equipped with 16 GB RAM. Two case studies using a noisy and non-convex function is discuss in this work. To verify the usefulness of the proposed EB-PSO in the field of analog circuit sizing having a multi-dimensional objective function with stringent design constraints, this work includes an additional case study involving a two-stage operational amplifier (op-amp). The validation of the obtained design parameters and circuit simulations for the third case study are done in Cadence-Virtuoso environment using 45 nm CMOS technology with 1.8 V supply. The parameters selected for implementing the proposed EB-PSO algorithm in all case studies are outlined in the Table 1. The literature [25] reports that the sum of c_1 and c_2 is stated to be greater than or equal to 4. Thus, $c_1 = 3$ and $c_2 = 1$ are selected for this work. The supply voltage, technology, and process parameters considered for the third case study are listed in Table 2. Here, $K_p = \mu_p C_{ox}$ and $K_n = \mu_n C_{ox}$. The value of $\zeta_{1,min}$ and $\zeta_{2,max}$ is chosen to be 0.1 and 1 respectively in this work.

A. CASE STUDY 1: NOISY FUNCTION

Consider the objective function $f_2(x)$ as defined in eq. (8).

$$\begin{aligned} &\text{Minimize} \quad f_2(x) = (x + rand(len(x)) * 0.3)^2 \\ &s.t. \quad -5 \leq x \leq 5 \end{aligned} \tag{8}$$

The function $f_2(x)$ is minimized using conventional PSO as well as proposed EB-PSO algorithm. It achieves the minimum value of $5.3 * 10^{-8}$ for $x = -0.06$. The errors ε_1 and ε_2 were evaluated with iterations count for both the considered PSO algorithms. The results are plotted in Fig. 7. It is noteworthy that the proposed EB-PSO algorithm takes 62 iterations for ε_1 to settle whereas the conventional PSO takes 79 iterations. Also, the error ε_2 settles in 35 iterations whereas the conventional PSO does not settle within the considered bound ($\alpha_2 = 0.3$). It should be noted that ε_2 settles before ε_1 as discussed in Section III.

B. CASE STUDY 2: NON-CONVEX UNI-MODAL FUNCTION

Another case study of non-convex uni-modal function is considered in this work, whose objective function $f_3(x)$ is

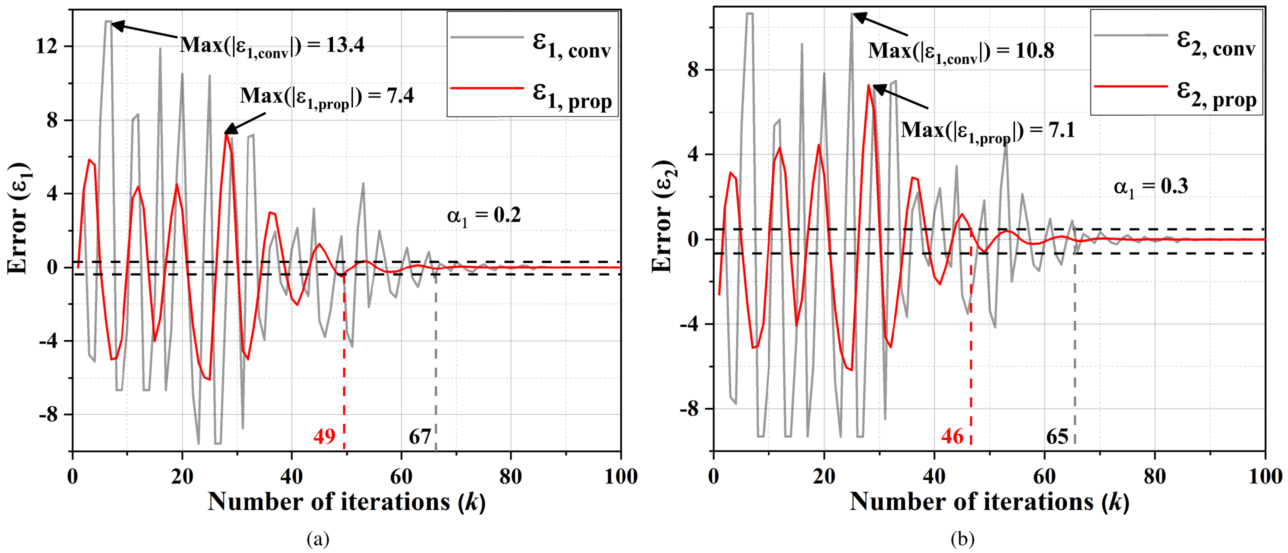


FIGURE 8. Variation of local error (ε_1) and global error (ε_2) with iterations count for the objective function $f_3(x)$.

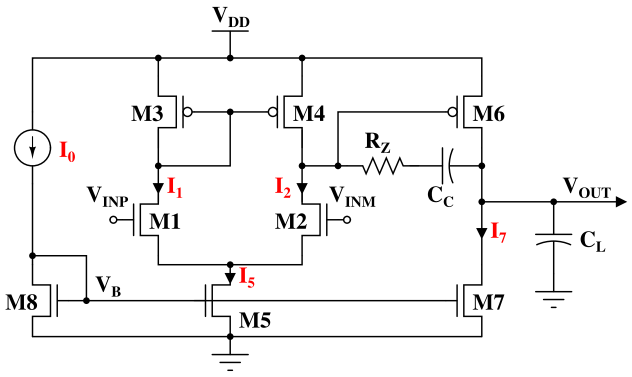


FIGURE 9. Case study 3: Schematic illustrating a two-stage op-amp.

given by,

$$\begin{aligned} &\text{Minimize} \quad f_3(x) = -(x + \sin x) * \exp(-x^2) \\ &s.t. \quad -10 \leq x \leq 10 \end{aligned} \tag{9}$$

The function achieves a minimum value of −0.824 for $x = 0.68$ with a proposed EB-PSO. The errors ε_1 and ε_2 evaluated using conventional and proposed EB-PSO are plotted in Fig. 8. It can be observed that the proposed EB-PSO takes 49 and 46 iterations for ε_1 and ε_2 to settle, whereas conventional PSO takes 67 and 65 iterations for ε_1 and ε_2 to settle. Thus, it is noticed that proposed EB-PSO converges approximately 27% faster than the conventional PSO.

C. CASE STUDY 3: TWO-STAGE OP-AMP

In assessing the effectiveness of the EB-PSO in analog circuit sizing, a case study is undertaken that involves a two-stage operational amplifiern (op-amp) with RC compensation. The schematic diagram of the two-stage op-amp is depicted in Fig. 9. A load capacitance of 1 pF is considered. The design steps [27], meeting specified constraints, are summarized in Table 3. The problem formulation encompasses 10 design variables, and their search ranges are detailed in Table 4. The primary objective of this case study is to minimize the total power consumption in two-stage op-amp.

The formulation of the optimization problem is expressed as follows:

$$\begin{aligned} &\text{Minimize} \quad \text{Power} \\ &s.t. \quad \text{DC Gain} \geq 60\,\text{dB} \\ &\quad\quad \text{PM} \geq 60^\circ \\ &\quad\quad \text{GBW} \geq 15\,\text{MHz} \\ &\quad\quad \text{SR} \geq 20\,\text{V}/\mu\text{s} \\ &\quad\quad \text{Output swing} \geq 1.2\,\text{V} \\ &\quad\quad \text{V}_{\text{IN}}\ (\text{max}) = 1.6\,\text{V} \\ &\quad\quad \text{V}_{\text{IN}}\ (\text{min}) = 0.6\,\text{V} \end{aligned} \tag{10}$$

In this particular study, the objective is to ensure that all transistors operate in the saturation region for enhanced gain. Consequently, the saturation margins are maintained well above 50 mV. This adds to a total of 15 specifications in the design considerations. The cost function for the proposed along with the considered traditional algorithms is plotted in Fig. 10. It is observed that the proposed EB-PSO algorithm converges in 11 iterations, whereas it takes 23, 29, and 41 iterations to converge for conventional GA, DE, and PSO algorithms respectively. The running time for each considered algorithms are tabulated in Table 5. Notably, the Genetic Algorithm (GA) exhibits a faster convergence time. However, it is worth highlighting that the optimal solution achieved with the EB-PSO algorithm is significantly closer to the desired specifications in comparison to GA. The optimized design, acquired through the proposed EB-PSO algorithm, is detailed in Table 6. To affirm the precision of the optimal design parameters, diverse analyses including transient

TABLE 3. Steps in designing Case Study 3, focusing on a two-stage op-amp [27].

Step	Relation	Design Equation
1	Phase Margin (PM) $\geq 60^o$	$g_{m6} \geq 10 g_{m1} \implies C_c \geq 0.22 C_L$
2	Slew rate	SR = $\frac{I_5}{C_c}$
3	Input transconductance	$g_{m1,2} = 2\pi f_{ugb} C_c$
4	$f_{ugb} = \frac{g_{m1,2}}{C_L}$	$\left(\frac{W}{L}\right)_{1,2} = \frac{g_{m1,2}^2}{\mu_n C_{ox} * I_5}$
5	ICMR (max)	$\left(\frac{W}{L}\right)_{3,4} = \frac{I_5}{\mu_p C_{ox}\left[V_{DD} - V_{IN}(max) - \lvert V_{thp}\rvert + V_{thn}\right]^2}$
6	ICMR (min)	$V_{D,sat} \geq V_{IN}(min) - \sqrt{\frac{I_5}{\mu_n C_{ox}\left(\frac{W}{L}\right)_{1,2}}} - V_{thn}$
7	Width of tail transistor	$\left(\frac{W}{L}\right)_5 = \frac{2I_5}{\mu_n C_{ox}\left(V_{D,sat}\right)^2}$
8	Vout (max)	$\left(\frac{W}{L}\right)_6 = \left(\frac{g_{m6}}{g_{m4}}\right) * I_4$ where $g_{m4} = \sqrt{\mu_p C_{ox}\left(\frac{W}{L}\right)_4 I_5}$
9	Vout (min)	$\left(\frac{W}{L}\right)_7 = \left(\frac{I_6}{I_5}\right) * \left(\frac{W}{L}\right)_5$
10	Bias current	$\left(\frac{W}{L}\right)_8 = \left(\frac{I_0}{I_5}\right) * \left(\frac{W}{L}\right)_5$
11	Pole-splitting	$R_z = \frac{C_L + C_c}{g_{m6} * C_c}$
12	Differential voltage gain	$A_{vd} = \frac{2 * g_{m1,2} * g_{m6}}{I_5 I_6 \left(\lambda_n + \lambda_p\right)^2}$
13	Power dissipation	$P = \left(I_0 + I_5 + I_7\right) * V_{DD}$

TABLE 4. Parameters for the design and their respective ranges for a two-stage op-amp.

Parameter	LB	UB	Parameter	LB	UB
W_1	0.12 μm	100 μm	W_6	0.12 μm	100 μm
W_2	0.12 μm	100 μm	W_7	0.12 μm	100 μm
W_3	0.12 μm	100 μm	W_8	0.12 μm	100 μm
W_4	0.12 μm	100 μm	C_c	0.1 pF	10 pF
W_5	0.12 μm	100 μm	R_z	0.1 kΩ	100 kΩ

W: Widths; LB: Lower Bound; UB: Upper Bound.

TABLE 5. Running time for different considered algorithms in sec.

Algorithms	Running Time
Conventional PSO	8.35
DE	12.24
GA	4.38
EB-PSO (This Work)	8.96

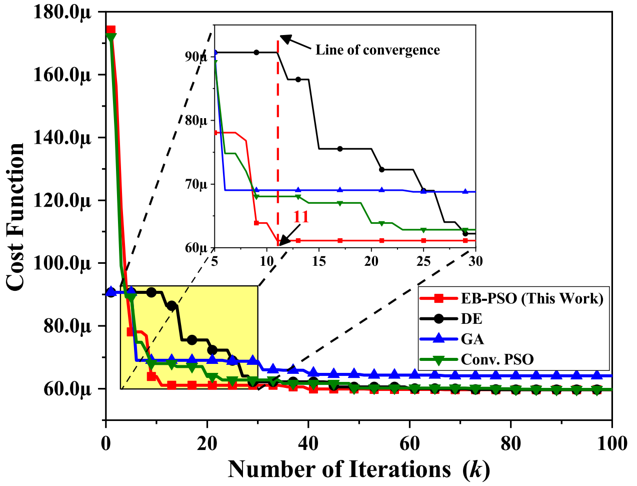


FIGURE 10. Variation in cost function of a two-stage op-amp concerning the iteration count for various evolutionary algorithms.

simulations, DC simulations, and frequency response evaluations were carried out.

1) DC ANALYSIS

The determination of the Input Common-Mode Range (ICMR) involves setting up the op-amp in the form of a non-inverting unity gain buffer. In this configuration, a continuous input spanning from 0 to V_{DD} is applied to the op-amp. The output voltage is graphed as illustrated in Fig. 11. It's important to highlight that the output maintains linearity within a defined range of input voltages. Consequently, V_{IN}(min) and V_{IN}(max) are determined to be 0.38 V and 1.57 V, respectively.

2) TRANSIENT ANALYSIS

A time-domain analysis is conducted with a step signal as input. The slew-rate is determined by measuring the slope of the output signal, as shown in Fig. 12. Utilizing the EB-PSO algorithm, a negative and positive slew-rate of 18.6 V/μs and 30 V/μs respectively are achieved. Additionally, the

TABLE 6. Parameters for the design of two-stage op-amp.

Design Parameters	Conventional PSO	DE	GA	SHPSO [37]	GPSOA [36]	SSA [38]	CHIMP [35]	This Work	Units
$\mathbf{I_0}$	1.155	1.59	0.136	1.2	56	0.92	1	1.18	μA
$\mathbf{(W/L)_{1\text{-}2}}$	0.54/2	0.585/2	0.368/2	0.29/1	0.46/1	0.27/1	0.92/1	0.56/2	$\mu m/\mu m$
$\mathbf{(W/L)_{3\text{-}4}}$	0.425/2	0.46/2	0.672/2	0.21/1	2..2/1	0.23/1	5.5/1	0.44/2	$\mu m/\mu m$
$\mathbf{(W/L)_5}$	1.52/2	2/2	20.89/2	0.75/1	13.7/1	1.05/1	15.7/1	1.76/2	$\mu m/\mu m$
$\mathbf{(W/L)_6}$	5.51/2	5.3/2	2.122/2	2.64/1	9.85/1	2.65/1	1.6/1	5.3/2	$\mu m/\mu m$
$\mathbf{(W/L)_7}$	9.856/2	11.54/2	33.024/2	4.86/1	30.85/1	5.95/1	1.82/1	10.5/2	$\mu m/\mu m$
$\mathbf{(W/L)_8}$	0.96/2	0.496/2	0.404/2	0.23/1	24/1	0.2/1	0.13/1	0.4/2	$\mu m/\mu m$
$\mathbf{C_c}$	0.22	0.22	0.22	0.22	0.84	0.22	1	0.22	pF
$\mathbf{R_z}$	25.718	26.604	66.804	26.8	2.7	26.7	15	26.734	kΩ

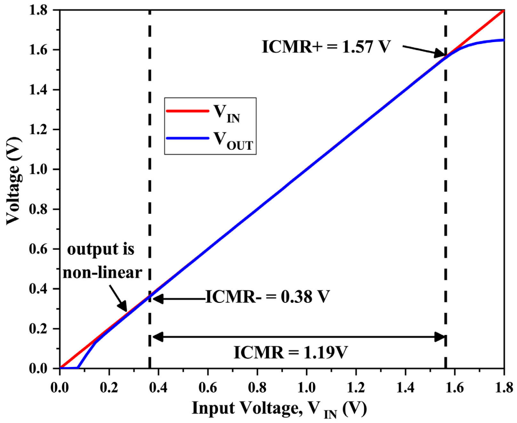


FIGURE 11. Variation in the output voltage of a two-stage op-amp in relation to the input common-mode range.

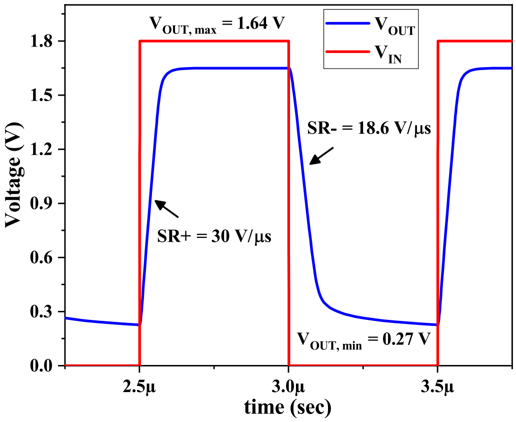


FIGURE 12. Unit step response of the two-stage op-amp in unity feedback configuration.

minimum and maximum output voltage turns out to be 0.27 V and 1.67 V, respectively.

3) FREQUENCY RESPONSE

In Fig. 13, a frequency response is depicted by analyzing a differential signal at the input. The designed system achieves a DC gain of 59.88 dB and a phase margin (PM) of 76.17°.

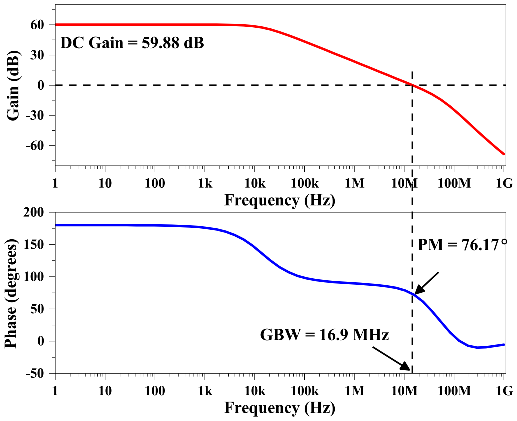


FIGURE 13. Frequency response of a two-stage op-amp.

Furthermore, the unity-gain bandwidth (f_{ugb}) is measured at 16.9 MHz, meeting the specified criteria.

For a comprehensive comparison, traditional evolutionary algorithms such as conventional PSO, differential evolution (DE), and genetic algorithm (GA) were employed alongside the proposed EB-PSO algorithm. The outcomes are presented in Table 7. Notably, the proposed algorithm demonstrates a higher convergence rate, effectively meeting the desired specifications. It is crucial to observe that both the proposed algorithm and DE successfully adhere to the unity gain bandwidth constraints, while conventional PSO and GA fall short in meeting the requirements. However, DE achieves the specifications at the expense of increased power consumption and convergence time. Moreover, it's worth noting that both GSA and CHIMP exhibit higher computational power requirements compared to the proposed EB-PSO algorithm. In contrast, EB-PSO demonstrates a closer alignment with design specifications when compared to HPSO and SSA optimizers. This suggests that EB-PSO not only offers a more efficient computational performance but also tends to yield solutions that are more closely matched to the specified design criteria than its counterparts.

TABLE 7. Comparative evaluation of two-stage op-amp performance using various evolutionary algorithms.

Parameters	**Expected**	**Conventional**	**DE**	**GA**	**SHPSO**	**GPSOA**	**SSA**	**CHIMP**	**This Work**
	Criteria	**PSO**			[37]	[36]	[38]	[35]	**(EB-PSO)**
PM (°)	> 60	77.02	73.7	71.4	77	86	78	60	76.17
Gain (dB)	60	62.5	57.19	52.5	56	53.4	55.4	53.1	59.88
SR ($\mathrm{V}/\mu\mathrm{s}$)	> 10	10.5	26	7.7	16.2	35	18.4	37	18.6
$\mathbf{V_{IN}}$ **(min) (V)**	0.6	0.35	0.42	0.45	0.3	0.42	0.35	0.4	0.38
$\mathbf{V_{IN}}$ **(max) (V)**	1.6	1.56	1.44	1.52	1.66	1.68	1.65	1.7	1.57
GBW (MHz)	> 15	9.94	21.5	14.78	10.38	10	10.35	13.5	16.9
Delay (ns)	< 50	56.2	18	18.7	44.2	19	46	11	32.9
Power (μW)	< 70	33	85.3	25.15	40	294	42	128	62.43
Total MOS area ($\mu\mathrm{m}^2$)	< 50	37.706	20.272	114.96	9.48	88.87	10.86	32.07	35.122

V. CONCLUSION

An Error-Bound Particle Swarm Optimization (EB-PSO) algorithm was proposed for analog circuit sizing to improve the convergence rate in this work. An optimal solution was found in a constrained design environment. The parameters (ζ_1 and ζ_2) were introduced in the velocity update equation. An exponential behavior bounded between ($\zeta_{1,min}$, $\zeta_{2,min}$) and ($\zeta_{1,max}$, $\zeta_{2,max}$) was proposed with respect to the number of iterations for both the parameters. The local error (ε_1) and global error (ε_2) were calculated iteratively. Initially, ε_2 is minimized leading to global exploration search. It is then followed by local exploitation by minimizing ε_1. To verify the effectiveness of the proposed EB-PSO algorithm in the field of analog circuit sizing, a case study of two-stage op-amp was presented. The optimal solution was verified by simulation results using Cadence-Virtuoso environment. The results showed that the optimal solution meets all the desired specifications. Furthermore, in comparison to other traditional algorithms under consideration, the proposed algorithm has demonstrated the highest convergence rate of 11 iterations in case of two-stage op-amp, which is 52%, 62%, and 73% faster when compared to GA, DE and conventional PSO algorithm.

REFERENCES

[1] T. Sennewald, F. Linke, J. Reck, and D. Westermann, "Parameter optimization of differential evolution and particle swarm optimization in the context of optimal power flow," in *Proc. IEEE PES Innov. Smart Grid Technol. Eur. (ISGT-Europe)*, Oct. 2020, pp. 1045–1049, doi: 10.1109/ISGT-Europe47291.2020.9248839.

[2] Y. F. Yiu, J. Du, and R. Mahapatra, "Evolutionary heuristic a search: Heuristic function optimization via genetic algorithm," in *Proc. IEEE 1st Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Sep. 2018, pp. 25–32, doi: 10.1109/AIKE.2018.00012.

[3] V. Bulitko, "Evolving initial heuristic functions for agent-centered heuristic search," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 534–541, doi: 10.1109/CoG47356.2020.9231637.

[4] Q.-Z. Xiao, J. Zhong, L. Feng, L. Luo, and J. Lv, "A cooperative coevolution hyper-heuristic framework for workflow scheduling problem," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 150–163, Jan. 2022, doi: 10.1109/TSC.2019.2923912.

[5] M. Esteve, J. Javier Rodríguez-Sala, J. J. López-Espín, and J. Aparicio, "Heuristic and backtracking algorithms for improving the performance of efficiency analysis trees," *IEEE Access*, vol. 9, pp. 17421–17428, 2021, doi: 10.1109/ACCESS.2021.3054006.

[6] M. Greco and C. Hernandez, "Heuristic function to solve the generalized covering TSP with artificial intelligence search," in *Proc. 39th Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Nov. 2020, pp. 1–8, doi: 10.1109/SCCC51225.2020.9281156.

[7] O. Buffet, J. Dibangoye, A. Saffidine, and V. Thomas, "Heuristic search value iteration for zero-sum stochastic games," *IEEE Trans. Games*, vol. 13, no. 3, pp. 239–248, Sep. 2021, doi: 10.1109/TG.2020.3005214.

[8] S. A. G. van der Stockt, A. P. Engelbrecht, and C. W. Cleghorn, "Heuristic space diversity measures for population-based hyper-heuristics," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–9, doi: 10.1109/CEC48606.2020.9185719.

[9] I. M. Ismail and N. N. Agwu, "Influence of heuristic functions on real-time heuristic search methods," in *Proc. 14th Int. Conf. Electron. Comput. Comput. (ICECCO)*, Nov. 2018, pp. 206–212, doi: 10.1109/ICECCO.2018.8634782.

[10] C. Huang, Y. Li, and X. Yao, "A survey of automatic parameter tuning methods for metaheuristics," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 201–216, Apr. 2020, doi: 10.1109/TEVC.2019.2921598.

[11] X. Wang, K. Zhang, J. Wang, and Y. Jin, "An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 859–871, Oct. 2022, doi: 10.1109/TEVC.2021.3111209.

[12] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, Feb. 2021, doi: 10.1109/TEVC.2020.3004012.

[13] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 941–960, Oct. 2022, doi: 10.1109/TEVC.2021.3139437.

[14] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, and C. Yue, "An evolutionary multitasking optimization framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 263–277, Apr. 2022, doi: 10.1109/TEVC.2022.3145582.

[15] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 201–221, Apr. 2023, doi: 10.1109/TEVC.2022.3155533.

[16] M. N. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization—Part I," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 802–822, Oct. 2022, doi: 10.1109/TEVC.2021.3130838.

[17] A. Kazikova, M. Pluhacek, and R. Senkerik, "How does the number of objective function evaluations impact our understanding of metaheuristics behavior?" *IEEE Access*, vol. 9, pp. 44032–44048, 2021, doi: 10.1109/ACCESS.2021.3066135.

[18] Q. Li, Y. Bai, and W. Gao, "Improved initialization method for metaheuristic algorithms: A novel search space view," *IEEE Access*, vol. 9, pp. 121366–121384, 2021, doi: 10.1109/ACCESS.2021.3073480.

[19] J. Xu and L. Xu, "Optimal stochastic process optimizer: A new metaheuristic algorithm with adaptive exploration-exploitation property," *IEEE Access*, vol. 9, pp. 108640–108664, 2021, doi: 10.1109/ACCESS.2021.3101939.

[20] F. Almeida, D. Giménez, J. J. López-Espín, and M. Pérez-Pérez, ''Parameterized schemes of metaheuristics: Basic ideas and applications with genetic algorithms, scatter search, and GRASP,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 570–586, May 2013, doi: 10.1109/TSMCA.2012.2217322.

[21] A. A. Muazu, A. S. Hashim, and A. Sarlan, ''Review of nature inspired metaheuristic algorithm selection for combinatorial t-way testing,'' *IEEE Access*, vol. 10, pp. 27404–27431, 2022, doi: 10.1109/ACCESS.2022.3157400.

[22] S. Ghosh, B. P. De, R. Kar, D. Mandal, and A. K. Mal, ''Optimal design of complementary metal-oxide-semiconductor analogue circuits: An evolutionary approach,'' *Comput. Electr. Eng.*, vol. 80, Dec. 2019, Art. no. 106485.

[23] J. Kennedy and R. Eberhart, ''Particle swarm optimization,'' in *Proc. Int. Conf. Neural Netw.*, Dec. 1995, pp. 1942–1948, doi: 10.1109/icnn.1995.488968.

[24] A. Banks, J. Vincent, and C. Anyakoha, ''A review of particle swarm optimization—Part I: Background and development,'' *Natural Comput.*, vol. 6, no. 4, pp. 467–484, Dec. 2007.

[25] D. Bratton and J. Kennedy, ''Defining a standard for particle swarm optimization,'' in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 120–127, doi: 10.1109/SIS.2007.368035.

[26] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, ''Particle swarm optimization: A comprehensive survey,'' *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[27] 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.

[28] P. Das and B. Jajodia, ''Design automation of two-stage operational amplifier using multi-objective genetic algorithm and SPICE framework,'' in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Jul. 2022, pp. 166–170, doi: 10.1109/ICICT54344.2022.9850461.

[29] R. Zhou, P. Poechmueller, and Y. Wang, ''An analog circuit design and optimization system with rule-guided genetic algorithm,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 12, pp. 5182–5192, Dec. 2022, doi: 10.1109/TCAD.2022.3166637.

[30] S. Yin, W. Zhang, W. Hu, Z. Wang, R. Wang, J. Zhang, and Y. Wang, ''An efficient reference-point based surrogate-assisted multi-objective differential evolution for analog/RF circuit synthesis,'' in *Proc. IEEE Int. Symp. Radio-Frequency Integr. Technol. (RFIT)*, Aug. 2021, pp. 1–3, doi: 10.1109/RFIT52905.2021.9565309.

[31] S. Yin, W. Hu, R. Wang, Z. Wang, J. Zhang, and Y. Wang, ''Surrogate-assisted multi-objective differential evolution based on Gaussian process for analog circuit synthesis,'' in *Proc. SMACD/PRIME Int. Conf. SMACD 16th Conf.*, Jul. 2021, pp. 1–4.

[32] J. Li, J. Zhang, C. Jiang, and M. Zhou, ''Composite particle swarm optimizer with historical memory for function optimization,'' *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015, doi: 10.1109/TCYB.2015.2424836.

[33] B. Biswal, P. K. Dash, and B. K. Panigrahi, ''Power quality disturbance classification using fuzzy C-means algorithm and adaptive particle swarm optimization,'' *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 212–220, Jan. 2009, doi: 10.1109/TIE.2008.928111.

[34] R. Rashid and N. Nambath, ''Hybrid particle swarm optimization algorithm for area minimization in 65 nm technology,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5, doi: 10.1109/ISCAS51556.2021.9401139.

[35] C. L. Kumari, V. K. Kamboj, S. K. Bath, S. L. Tripathi, M. Khatri, and S. Sehgal, ''A boosted chimp optimizer for numerical and engineering design optimization challenges,'' *Eng. Comput.*, vol. 39, no. 4, pp. 2463–2514, Aug. 2023, doi: 10.1007/s00366-021-01591-5.

[36] S. Jiang, C. Zhang, W. Wu, and S. Chen, ''Combined economic and emission dispatch problem of wind-thermal power system using gravitational particle swarm optimization algorithm,'' *Math. Problems Eng.*, vol. 2019, pp. 1–19, Nov. 2019, doi: 10.1155/2019/5679361.

[37] S. Jiang, C. Zhang, and S. Chen, ''Sequential hybrid particle swarm optimization and gravitational search algorithm with dependent random coefficients,'' *Math. Problems Eng.*, vol. 2020, pp. 1–17, Apr. 2020, doi: 10.1155/2020/1957812.

[38] C. Zhang, L. Luo, Z. Yang, S. Zhao, Y. He, X. Wang, and H. Wang, ''Battery SOH estimation method based on gradual decreasing current, double correlation analysis and GRU,'' *Green Energy Intell. Transp.*, vol. 2, no. 5, Oct. 2023, Art. no. 100108, doi: 10.1016/j.geits.2023.100108.

[39] Y. Shi and R. Eberhart, ''A modified particle swarm optimizer,'' in *Proc. IEEE Int. Conf. Evol. Comput., IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.

K. G. SHREEHARSHA received the B.E. degree from the Vivekananda College of Engineering and Technology, Puttur, under Visvesvaraya Technological University (VTU) Belagavi, India, in 2008, and the M.Tech. degree in digital electronics and communication from the NMAM Institute of Technology, Nitte, under VTU Belagavi, in 2010. He is currently an Assistant Professor with the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India.

R. K. SIDDHARTH (Member, IEEE) was born in Gujarat, India, in 1991. He received the B.E. degree from the Government Engineering College, Patan, India, in 2013, and the M.Tech. and Ph.D. degrees in VLSI from the National Institute of Technology (NIT) Goa, India, in 2016 and 2021, respectively. He was a Research Associate with NIT Goa, until July 2023. He is currently an Assistant Professor with the Department of Mechatronics, Parul University, Gujarat, India. He has published more than 25 research papers in journals/conferences and filed several patents. His research interests include analog and mixed signal design, memory design, and data converters.

M. H. VASANTHA (Member, IEEE) received the B.Tech. degree from the NMAM Institute of Technology, Nitte, under Mangalore University, India, in 1998, the M.Tech. degree in microelectronics and VLSI design from IIT Madras, Chennai, India, in 2003, and the Ph.D. degree from the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, Surathkal, India, in 2014. He is currently an Associate Professor with the Department of ECE, National Institute of Technology (NIT) Goa, India.

Y. B. NITHIN KUMAR (Senior Member, IEEE) received the B.E. degree from Manipal Institute of Technology, Manipal, India, the M.Tech. degree from the National Institute of Technology Karnataka, Surathkal, India, and the Ph.D. degree from IIT Kharagpur, Kharagpur, India. He was a Visiting Researcher with the University of Pavia, Italy. He is currently an Associate Professor with the National Institute of Technology Goa, India. He has published more than 75 research papers in journals/conferences and filed several patents. He was a recipient of the Young Indian Researcher Award from the Government of Italy, from 2007 and 2010, the Young Scientists in Engineering Sciences from DST, Government of India, in 2015, and the Young Faculty Research Fellowship (YFRF) of Visvesvaraya Ph.D. Programme of Ministry of Electronics and Information Technology, MeitY, Government of India, in 2018.