Improvement in Sizing Constrained Analog IC via Ts-CPD Algorithm

Pedro Lagos-Eulogio , Pedro Miranda-Romagnoli * , Juan Carlos Seck-Tuoh-Mora * , Norberto Hernández-Romero

Posted Date: 10 October 2023

doi: 10.20944/preprints202310.0597.v1

Keywords: cellular particle swarm optimization (CPSO); constrained optimization; circuit sizing tool; particle swarm optimization

Article

Improvement in Sizing Constrained Analog IC via Ts-CPD Algorithm

Pedro Lagos-Eulogio [1,†], Pedro Miranda-Romagnoli [1,*,†], Juan Seck-Tuoh-Mora [2,*,†] and Norberto Hernández-Romero [2,†]

[1] Área Académica de Matemáticas y Física, Instituto de Ciencias Básicas e Ingeniería, Universidad Autónoma del Estado de Hidalgo, Carr. Pachuca-Tulancingo km. 4.5, Pachuca 42184, Hidalgo, Mexico; plagos@uaeh.edu.mx (P.L.-E.); pmiranda@uaeh.edu.mx (P.M.-R.)

[2] Área Académica de Ingeniería, Instituto de Ciencias Básicas e Ingeniería, Universidad Autónoma del Estado de Hidalgo, Carr. Pachuca-Tulancingo km. 4.5, Pachuca 42184, Hidalgo, Mexico; jseck@uaeh.edu.mx (J.C.S.-T.-M.); nhromero@uaeh.edu.mx (N.H.-R.)

* Correspondence: pmiranda@uaeh.edu.mx; (P.M.-R.); jseck@uaeh.edu.mx; (J. C. S.-T.-M.)

† These authors contributed equally to this work.

Abstract: In this work, we propose a variation of the cellular particle swarm optimization algorithm with differential evolution hybridization (CPSO-DE), to include constrained optimization in it, named Ts-CPD. It is implemented as a kernel of electronic design automation (EDA) tool capable of sizing circuit components considering a single-objective design with restrictions and constraints. The aim is to improve the optimization solutions in the sizing of analog circuits. To evaluate our proposal's performance, we present the design of three analog circuits: a deferential amplifier, a two-stage operational amplifier, and a folded cascode operational transconductance amplifier. Numerical simulation results indicate that Ts-CPD can find better solutions, in terms of the design objective and the accomplishment of constraints, than those reported in previous works.

Keywords: cellular particle swarm optimization (CPSO); constrained optimization; circuit sizing tool; particle swarm optimization

1. Introduction

In recent years, analog circuit design has received much attention, particularly those with Very Large Scale of Integration (VLSI), because optimization is a process that involves many conflicting constraints and a wide range of parameters [1]. Therefore, it is necessary to develop more robust Computer-Aided Design (CAD) and Electronic Design Automation (EDA) tools, which increase productivity and quality and minimize design costs [2].

The design of analog circuits comprises three major stages: selecting a topology, sizing components, and layout extraction [3]. In the case of sizing, it is possible to use the experience when the circuits are small, but manual circuit-sizing in analog design is a time-consuming process [4]. When the circuit grows, it is impossible to size the components solely by experience; thus, mathematical tools are necessary to optimize the circuits. Similarly, component selection is critical in achieving a targeted performance and quality level [5].

The complexity when manually implementing an analog project is usually weeks or months. CAD and EDA tools are used to improve the design process; today's analog design environment is made of CAD tools for editing, evaluation, and design verification of analog integrated circuits, for example, HSPICE, SMASH, and CADENCE. Automatic analog circuit sizing based on optimization tools is divided into two main subclasses: equation-based and simulation-based circuit optimization. The first uses analytical design equations, while the second takes advantage of simulator circuits, e.g., SPICE, to provide accurate performance figures.

Critical points in optimization-based approaches are the optimization techniques and methods used to evaluate circuit performance. Different optimization techniques are used to dimension and

 doi:10.20944/preprints202310.0597.v1

optimize analog integrated circuits, both deterministic and stochastic. In equation-based methods, it is possible to use classical optimization methods. However, when using the circuit simulator, it is necessary to use stochastic heuristic optimization techniques [6].

Many optimization techniques and tools for automation design have been developed over time [7],[8]. Also, Geometric Programming was used to design a CMOS op-amp in [9]; the authors used the transistor models called GP0 and GP1, implemented in MATLAB but validated with the HSPICE level-1 models simulation. This method is robust, although it implies using equations for the circuit and each optimization parameter. This technique is updated in [10], where the validation is made using the BSIM3v1 model.

In addition, the fuzzy logic has been used for the circuit design as in [11,12], or in [13], where a multi-objective design is presented, while in [14], a tool for analog synthesis is introduced. In [15], a Neuro-Fuzzy method for analog circuit design is presented; it is of easy implementation, natural understanding, and better performance than static methods of fuzzy optimization; however, it still needs the human experience in the particular circuit to be designed. In [16], the application of an innovative algorithm of the type Customized Genetic Algorithm (CAG) is reported. Its purpose is to improve the optimization process of analog CMOS ICs.

More recently, evolutionary algorithms have been successfully applied to component value selection for analog active filters [17], [18] and to the analog integrated circuits design as in [19], where the sizing is achieved using a Particle Swarm Optimization (PSO) algorithm implemented in MATLAB and the results verified at the end with SPICE. In [20], a CMOS differential amplifier and a two stages CMOS op-amp are optimized to occupy the minimal possible area by the circuits and to improve their performances using the gravitational search algorithm in combination with the particle swarm optimization (GSA-PSO). The design is formulated as an optimization problem with a single objective function, although certain manual tuning is necessary to resolve conflicts with either design or performance parameters when using this method. In the work [21], a crazy PSO (CRPSO) is applied to improve the premature convergence to a local minimum of the PSO; the application optimizes the minimization of the total MOS area of two amplifier configurations, a two-stage PMOS type operational amplifier, and an NMOS cascade code amplifier.

A good comparison of several evolutionary methods for the synthesis of analog circuits is presented in [22]; these are Artificial Bee Colony (ABC), PSO, and Chaotic Differential Evolution (CDE). The algorithms are implemented in MATLAB and interfaced with the WINSPICE circuits simulator. In [23], the authors present their tool for the automation design of analog circuits based on the use of a Genetic Algorithm (GA) modified; this is a multi-objective design for CMOS op-amps. Another evolutionary algorithm used for automation design is the New Hybrid Shuffled Frog Leaping Algorithm (NHSFL) implemented in MATLAB linked with the HSPICE circuits simulator [24]. It was tested with two examples of design, but the method can be extended to the general op-amp design according to the authors.

Heuristic techniques are necessary to solve problems with many design criteria [25]. Although they do not guarantee finding the optimal solution exactly, they provide an acceptable approximation to it in an acceptable computation time [26]. Therefore, another challenge for sizing high-performance analog circuits with tight specifications is the need for a powerful enough optimization kernel for EDA tools to handle tighter specifications and improve optimization capability [27]. Different optimization kernels are currently used for EDA tools; among them, we can mention the kernels based on GA [23], PSO [28], ACO in [29], SA in [30], GSA in [20], NSGA-II in [31] and NSGA-II, MOPSO, and MOSA in [32].

Most heuristic methods used in the optimization kernels of the EDA tools are based on multi-objective optimization techniques [6,31] or use a restriction approach with a single objective and static penalty functions. This is due to their simplicity and easy implementation, making it a relatively reliable method [33]. Penalty functions penalize non-feasible solutions by adding a specific value to the objective function as an amount proportional to the violation of the restriction. Thus, the

 doi:10.20944/preprints202310.0597.v1

optimization problem is transformed into a restrictionless optimization problem. The main problem with this methodology is choosing the appropriate penalty factor for a particular problem; it is often a complicated task, but if an adequate factor is selected, a premature convergence can occur or solutions outside the feasible region can be obtained [34]. Another approach currently used in problems with restrictions is self-adaptive penalty functions, which significantly improve the results [35]. Unfortunately, many last-generation restricted optimization methods have yet to be introduced into EDA tools. Therefore, advanced restricted optimization methods should be applied to circuit dimensioning tools to address this challenge.

In recent years, algorithms based on cellular automata, such as CPSO [36], CPSO-DE [37], CCAA [38], and MmCAA [39], have shown excellent performance in solving global optimization problems, demonstrating a good balance between exploration and exploitation, as well as a good speed of convergence. Among them, the CPSO-DE has proven to be an excellent design method for identifying adaptive IIR systems due to the use of a differential evolution rule for the neighborhoods of cellular automata of the PSO that improves the balance between exploration and exploitation than the original version of the CPSO.

According to the previous observations, this document introduces the hybrid continuous optimization algorithm called CPSO-DE that incorporates cellular automata concepts to improve PSO exploitation capabilities with DE exploitability. The algorithm was tested on established benchmark functions (CEC 2005) [40] against 7 recently published algorithms for global optimization, yielded satisfactory results.

Additionally, Deb's rules were incorporated into the algorithm to address constrained optimization [41,42]; this algorithm is called Ts-CPD applied in a single design objective problem, for the sizing of analog circuits to improve their performance. The approach is used as the optimization core of an EDA tool to size CMOS analog circuits efficiently. In particular, we focus on diminishing the total component area as the objective. At the same time, other specifications, such as dc gain, bandwidth and power dissipation, are treated as constraints that guarantee good overall performance. The circuits chosen for testing our method are well known, which allows a comparison of results with other proposals. We implemented the optimization in Matlab while the circuit simulation was done in Ngspice. Both optimization and simulation parts are linked.

We compare our proposal with previously published works, including PSO variants such as Particle Swarm Optimization (PSO)[19], Genetic Algorithm (GA)[43], Harmony Search (HS)[44], Differential Evolution (DE)[44], Artificial Bee Colony (ABC)[44], Gravitational Search Algorithm PSO (GSA-PSO)[20], Geometric Programming (GP)[9] and Aging Leader and Challenger PSO (ALC-PSO)[1]. The results show that Ts-CPSO can find a better circuit design solution than the above-listed approaches. In addition, it shows a rapid convergence in all the studied cases.

Overall, the proposed CPSO-DE algorithm is easy to understand, performs exceptionally well for continuous optimization, and is modified with Deb's rules to define the Ts-CPD algorithm in order to tackle problems with multiple constraints, as demonstrated in the area optimization of CMOS analog circuits.

The rest of the paper is organized as follows: Section 2 gives a review of CPSO-DE, while the hybridization of CPSO-DE with constrained optimization is explained in Section 3. Section 4 describes three circuits in terms of their design variables and constraints. Section 5 validates the proposed Ts-CPD through three cases of study, contrasting the findings against results from previous works. Finally, this article is concluded in section 6.

2. Review of CPSO and CPSO-DE

2.1. Cellular particle swarm optimization

Particle Swarm Optimization (PSO) is one of the most frequently applied swarm intelligence-based algorithms for optimization tasks. PSO simulates the behavior of a bird flock, looking for an equilibrium

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

between exploration and exploitation of the current solutions. Particles in a d-dimensional search space are regarded as candidate solutions. We denote the i-th particle as,

$$X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}), \tag{1}$$

and its velocity as,

$$V_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,d}). \tag{2}$$

Each particle evolves in the search space, where $P_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,d})$ is the personal best position of the i-th particle so far and $G = (g_1, g_2, \ldots, g_d)$ is the global best position discovered by the swarm. At each time step t, both the velocity and position of each particle are updated to move it into a new position. Velocity and position are updated as follows:

$$\begin{aligned} V_i^{t+1} &= V_i^t + c_1 r_1 (P_i^t - X_i^t) + c_2 r_2 (G^t - X_i^t) \\ X_i^{t+1} &= X_i^t + V_i^t \end{aligned} \tag{3}$$

where c_1 and c_2 are two positive constants (cognitive and social factors), r_1 and r_2 are two uniform random numbers in $[0, 1]$. The fitness $h(X_i)$ of a particle gives its quality, that is, a better fitness value means a better particle.

Several papers have presented the adaptation, modification, and hybridization of PSO with other techniques to solve a huge variety of problems. Relevant surveys can be consulted in [45], [46], and [47].

One recent variant of PSO is the Cellular-PSO (CPSO). The idea behind CPSO is to apply convenient properties of cellular automata to enhance the performance of PSO [36]. In this reference, it is explained that there are two crucial factors in population-based optimization algorithms: communication mechanisms for the cooperation of the population and information inheriting for the self-adaption of each individual.

The concept of cellular automata (CA) was first proposed by Von Neumann and Ulam, and there are an increasing number of researchers using CA in physics, biology, social science, computer science, and so on [48–50]. The implementation of CA is very simple and intuitive, and it consists of interconnected cells that update their states synchronously at discrete time steps. Each cell follows a common local function defined by its neighbors. With this simple local interaction, CA is able to produce complex global behavior.

In this paper, we use the Cellular PSO Outer version (CPSO-outer). In this case, every particle improves its searching capability, generating new solutions not belonging to the swarm. The whole search space is considered the cell space, so every potential candidate solution in the search space can be a cell. Every particle in the swarm is a “smart-cell", defined by (1), able to construct its neighborhood by a local function, enhancing its searching capability.

The neighborhood function makes CPSO-outer differ from common PSO adopting static neighbors. Every particle X_i (or “smart-cell") in CPSO-outer generates a set of l neighbors $N_{i+1} \ldots N_{i+l}$ taking its current position and the global best position in order to realize a local search, following the next equation[36]:

$$N_{i+j} = \begin{cases} X_i^t + \frac{h(G)}{h(X_i^t)} R \circ V_i^t & h(X_i^t) \neq 0, \quad h(G) \geq 0 \\ X_i^t + \left| \frac{h(X_i^t)}{h(G)} \right| R \circ V_i^t & h(X_i^t) \neq 0, \quad h(G) < 0 \\ X_i^t + \left(\frac{e^{h(G)}}{e^{h(X_i^t)}} \right)^2 R \circ V_i^t & h(X_i^t) = 0, \quad h(G) \geq 0 \\ X_i^t + \left(\frac{e^{h(G)}}{e^{h(X_i^t)}} \right)^2 R \circ V_i^t & h(X_i^t) = 0, \quad h(G) < 0 \end{cases} \tag{4}$$

for $1 \leq j \leq l$. R is a vector composed of d uniform random numbers in $[-1, 1]$ to obtain random changes in the direction and distance of every new neighbor, and $\circ$ is the Hadamard product. The idea is that the search range of every particle would be negligible at early iterations when the difference of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

its fitness value with that of $h(G)$ is relatively significant. Then, when particles converge gradually to $h(G)$, a more extensive search range is used.

The neighbors generated by each particle are evaluated, and the neighbor with the best fitness value replaces the particle:

$$f(\phi) = min(h(X_i), h(N_{i+1}), \ldots, h(N_{i+l}))$$

$$X_\phi = \begin{cases} X_i & \text{if} \quad f(\phi) = h(X_i) \\ N_{i+j} & \text{if} \quad f(\phi) = h(N_{i+j}) \end{cases} \tag{5}$$

$$X_i^{t+1} = X_\phi^t.$$

This transition rule gives particles new information to explore the search space from an optimal local area to another optimal local area with better fitness value and enhance the diversity of the swarm. So CPSO-outer has more significant potential to search for the global optimum.

The CPSO has been applied and modified to solve a variety of theoretical and practical problems. For instance, in [51], CPSO is used to optimize a milling system. In [52], truss structures are optimized using variants of CPSO, and parameters controlling process planning are tuned by the application of CPSO [53]. Nevertheless, CPSO has not been implemented for sizing analog circuit components.

2.2. Hybrid cellular particle swarm optimization and differential evolution

Hybrid cellular particle swarm optimization and differential evolution (CPSO-DE) is a recent hybrid method that combines the features of PSO, CA, and DE[37], which is an incorporation of local differential search to the CPSO-outer algorithm.
The cellular automata elements used in the CPSO-DE algorithm are,

(a) configuration: (Q particles or smart-cells);
(b) cell space: the set of all cells;
(c) cell state: the particle's information at time t, $S_i^t = [X_i^t]$;
(d) neighborhood: $\Phi(i) = \{i + \delta_j\}, 1 \leq j \leq l$ (l is the neighborhood size). See Figure 1,
(e) transition rule: $S_i^{t+1} = \varphi(S_i^t \cup S_{\Phi(i)}^t)$.

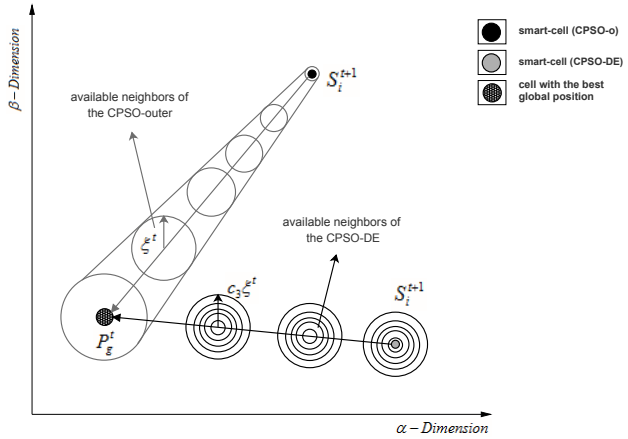


Figure 1. Neighborhood for CPSO-outer and CPSO-DE

In CPSO-DE, the i-th cell state S_i^t in the iteration t is updated using the PSO algorithm as follows:

$$V_i^{t+1} = w^t V_i^t + c_1 r_1 (P_i^t - S_i^t) + c_2 r_2 (P_g^t - S_i^t) \tag{6}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

$$S_i^{t+1} = S_i^t + V_i^{t+1} \tag{7}$$

where $i = 1, 2, \ldots, Q$ is the cell index and Q is the number of smart cells, c_1 and c_2 are the cognitive and social acceleration parameters respectively, r_1 and r_2 are two uniform distributed random numbers within $[0, 1]$, w is the inertial weight and decreases linearly. P_i is the previous personal best position, P_g is the global best position, and X_i and V_i are the current positions and velocity.
The operators used to determine each smart cell's neighborhoods are mutation and crossover. The mutation scheme "DE/rand/1"creates a new solution as follows:

$$O_{i,k}^t = S_{r_1}^t + c_3(S_{r_2}^t - S_{r_3}^t) \tag{8}$$

where $k = 1, 2, \ldots, l$ enumerates every neighbor, and l is the neighborhood size. The $r_1, r_2, r_3 \in \{1, 2, \ldots, Q\}$ are randomly chosen integers, distinct from each other and different from i. Factor c_3 is a real value between $[0, 2]$ for scaling the difference vector.
The crossover is an introduction to creating l trial vector $H_{i,k}$, combining the information of the current smart cell with each one of the l mutated vectors, as follows:

$$H_{i,j,k} = \begin{cases} O_{i,j,k}^t, & \text{if } r_{i,j} \leq C_r \text{ or } j = j_{rand}, \\ S_{i,j}^t, & \text{otherwise,} \end{cases} \tag{9}$$

where r_{ij} is a uniformly distributed random number within $[0, 1]$, $C_r \in [0, 1]$ is the crossover probability factor, and $j_{rand} \in \{1, 2, \ldots, D\}$ is a randomly chosen index, which ensures that $H_{i,k}$ copies at least one component from $O_{i,k}$. Finally, the transition rule is applied over the trial vectors to update the state of the current smart-cell:

$$S_i^{t+1}(P_\Phi) = \varphi(f(S_i^{t+1}), f(H_{i,1}), f(H_{i,2}), \ldots, f(H_{i,l})) \tag{10}$$

where the $f(.)$ are the fitness functions. In CPSO-outer, the neighborhood function $\Phi(i)$ generates random neighbors within radius ξ^t away from S_i^t according to its fitness value and the fitness of the best particle. Radius ξ^t is small when the smart-cell S_i^t is far from P_g^t, so the potential neighbors are close to S_i^t, and only when S_i^t converges to an equilibrium point, ξ^t would be a uniform random number in $[-1, 1]$. Therefore, the radius of neighborhoods in CPSO-outer increases when the particles stabilize. Therefore, the best results are obtained up to the last iterations.
On the other hand, CPSO-DE generates a random neighbor within radius $\xi^t = c_3(S_{r2}^t - S_{r3}^t)$. Thus, the radius of neighborhoods depends on the distribution and the improved information of the swarm as iteration passes, not just from the difference with the best global position. Thus, S_i^t is more likely to obtain better neighbors in any iteration time.

3. Tournament-selection CPD

The use of cellular automata in heuristic algorithms has been shown to be efficient, more specifically, in the use of adaptive IIR filters through the hybridization of the CPSO and DE algorithms that use a rule based on the use of neighborhoods. However, the CPSO-DE algorithm for the problem with restrictions on the sizing of CMOS circuits has not yet been reported in the literature, hence the motivation for this work.

In this section, we explain the parts that comprise the proposed Ts-CPD algorithm. First, we describe the optimization problem to be solved, which contemplates restrictions. Next, we explain how the initial values are selected, for our algorithm, using tournament selection (Ts), which is a variant of what Deb proposed [41]. We conclude this section by explaining the implementation of the Deb rule in the CPSO-DE, to build the new Ts-CPD algorithm.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

3.1. Constrained Optimization Problem

Many optimization problems in science and engineering implicate some constraints that the optimal solution must satisfy. For example, in a generic circuit, the optimization problem consists of finding optimal values of the design parameters. Then, a circuit design problem is usually written as a nonlinear programming (NLP) problem of the following type:

$$\begin{aligned} &\text{minimize } f(X) \\ &X \in \mathbb{R}^n \\ \text{Subject to:} \\ &|g_p| \geq spec_{g_p} \qquad p = 1 \cdots r \\ &h_q = spec_{h_q} \qquad q = 1 \cdots s \\ &x_{i,min} \leq x_i \leq x_{i,max} \quad i = 1 \cdots n \end{aligned} \tag{11}$$

In the above NLP problem, f is the cost function that maps the input space into the output one, $f : \mathbb{R}^n \to \mathbb{R}$, with $n = k + m$. There are two types of constraints, inequality constraints g_p that have to be major or minor than certain $spec_{g_p}$, and the equality constraints h_q, that has to be equal to the restriction $spec_{h_q}$. The ith variable varies in the range $[x_{i,min}, x_{i,max}]$.

The k independent variables and m dependent ones determine the circuit design represented in a single vector as,

$$X = (x_1, \ldots, x_k, x_{k+1}, \ldots, x_{k+m}). \tag{12}$$

The design variables and constraints for specific circuits studied in this paper are given in the Section 4.

3.2. Tournament-selection

As in the cost function f(X) of the optimization problem expressed in (11), the restrictions are not considered; we need a method that allows us to assess their contribution. In [41], Deb proposes a constraint handling method so that while the cost function is minimized, the constraints in the search for the minimum are considered. We will use Deb's method in this work, as explained below.

Let's say that the CPSO-DE algorithm has encountered two solutions for the problem (11), X_1 and X_2, according to the constrained optimization, solution X_1 is considered better if ([42]):

1. both solutions are feasible, but X_1 cost $\leq X_2$ cost; or,
2. X_1 is feasible but X_2 is not; or,
3. both solutions are unfeasible, but X_1 has less overall constraint violations than X_2.

These rules, implemented as Algorithm 1, are advantageous in finding a better solution for the circuit design, as will be shown in Section 5.

Algorithm 1 Tournament-Selection (X_1, X_2)

if X_1 is feasible **and** X_2 is feasible **then**
  if $f(X_1) < f(X_2)$ **then**
    return (X_1)
  else
    return (X_2)
  end if
else if constraints violation (X_1) < constraints violation (X_2) **then**
  return (X_1)
else
  return (X_2)
end if

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

3.3. Ts-CPD algorithm

This work proposes a new methodology that combines the CPSO-DE algorithm and Deb's rules for the problem of sizing CMOS analog circuits with constraints. The proposed algorithm, Ts-CPD, incorporates the tournament selection (*see* Algorithm 1) in the $\Psi()$ function. In this method, a new transition rule is proposed for Ts-CPD, which is applied to the trial vectors to update the state of the current smart-cell:

$$S_i^{t+1}(P_\Phi) = \Psi(\ldots\Psi(\Psi(S_i^{t+1}, H_{i,1}), H_{i,2}), \ldots, H_{i,l}) \tag{13}$$

The transition rule in (13) means that each cell in the neighborhood (including the same smart-cell) competes in a paired tournament (according to Deb criteria), and the winner is chosen to update the state of the smart-cell.

The proposed Ts-CPD method is described in Algorithm 2. First, the algorithm sets the control parameters Q, l, T, x_{min}, x_{max} y v_{max}. Next, the state (S) and velocity (V) are randomly initialized for each smart-cell. Then, each cell is evaluated, and its number of violated constraints is quantified. In line 9, Algorithm 1 is used to identify the best global position. The process halts according to the stopping criteria of iteration and convergence, according to line 10. Then, the cell state is updated using (6) and (7) in line 12. Later, the neighborhood of size l is generated for each smart-cell, using the DE method. Each neighbor is defined by the mutation and crossover rules in lines 14 and 15 using (8) and (9), respectively. The new transition rule inspired by Deb's rules and CA behavior, defined in 13, is applied in line 16 to determine the new cell state. Finally, the best local and global positions are updated in lines 18 and 19, respectively, using Algorithm 1. The process is repeated by each smart-cell and neighbor.

Algorithm 2 Ts-CPD

1: **//** Initialization**
2: Set the control parameters: Q, l, T, x_{min}, x_{max}, v_{max};
3: **for** $i = 1$ to Q **do**
4: Initialize $S_i \in (x_{min}, x_{max})$ randomly;
5: Initialize $V_i \in (-v_{max}, v_{max})$ randomly;
6: $P_i = S_i$;
7: **end for**
8: Evaluate each cell $f(S_i)$;
9: Identify the best global position (P_g): using Algorithm 1;
//* Loop**
10: **while** stopping criterion is not satisfied **do**
11: **for** $i = 1$ to Q **do**
12: **Update cell state:** using equations (6) and (7);
//*Generate l neighbors using DE method**
13: **for** $k = 1$ to l **do**
14: **Mutation rule:** using equation (8);
15: **Crossover rule:** using equation (9);
16: **New transition rule:** using equation (13);
17: **end for**
18: Identify the best local position (P_i): using Algorithm 1;
19: Identify the best global position (P_g): using Algorithm 1;
20: **end for**
21: **end while**

3.4. Performance of the Ts-CPD algorithm

To test the effectiveness of the Ts-CPD algorithm (without Deb's rules), we compared it to seven recently published global optimization algorithms, namely AOA [54], HHO [55], WSA [56], CCAA [38], MmCCAA [39], RECAA [57], and PO [58]. We used a total of 25 benchmark functions from CEC 2005 benchmark functions [40], which included five unimodal functions $(f_1, \ldots, f_5)$, seven basic multimodal functions $(f_6, \ldots, f_{12})$, two expanded multimodal functions (f_{13}, f_{14}), and 11 hybrid composition multimodal functions $(f_{15}, \ldots, f_{25})$.

We obtained the codes and parameters for these algorithms from the references cited in this study. This ensured that we used the same implementations as the original authors, making the comparison

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

more objective. Tables 1 and 2 present the average values and standard deviations of the objective function values obtained by each algorithm. We ran each algorithm independently 30 times.

In unimodal problems, the Ts-CPD algorithm showed excellent performance, ranking first among the eight algorithms in terms of average value. Moreover, it surpassed other algorithms in three cases based on standard deviation, highlighting its proficiency in information exploitation.

For the 20 multimodal and hybrid problems, Ts-CPD exhibited the highest average values in 12 instances. It also demonstrated its ability to explore and exploit simultaneously while maintaining robustness, achieving the best standard deviation values in four cases. Figure 2 shows some examples of the convergence curves for different test functions in 30 dimensions.

Table 1. Performance of metaheuristic algorithms compared with Ts-CPD on 30-dimensional unimodal problems.

Benchmark		Ts-CPD	AOA	HHO	WSA	CCAA	MmCAA	RECAA	PO
f_1	Avg	**1.08e+00**	3.54e+04	2.51e+03	7.42e+04	1.99e+04	2.23e+03	8.51e+02	1.88e+04
	Std	**2.93e+00**	5.35e+03	1.22e+03	5.97e+03	7.33e+03	6.25e+02	4.39e+02	7.28e+03
f_2	Avg	**1.60e+02**	4.11e+04	2.40e+04	1.11e+05	2.44e+04	4.23e+04	1.70e+04	1.92e+04
	Std	**1.68e+02**	5.24e+03	2.82e+03	3.02e+04	4.56e+03	6.21e+03	5.52e+03	4.97e+03
f_3	Avg	**2.60e+06**	5.23e+08	1.11e+08	1.00e+08	1.19e+08	7.82e+07	3.79e+07	1.68e+08
	Std	1.40e+06	1.74e+08	3.41e+07	**0.00e+00**	5.26e+07	2.80e+07	1.33e+07	1.05e+08
f_4	Avg	**9.37e+02**	4.83e+04	5.46e+04	1.11e+05	3.49e+04	5.26e+04	2.69e+04	2.89e+04
	Std	**4.80e+02**	7.16e+03	8.20e+03	2.13e+04	6.61e+03	7.99e+03	5.06e+03	6.62e+03
f_5	Avg	**5.78e+03**	2.97e+04	2.49e+04	4.32e+04	2.17e+04	1.41e+04	8.76e+03	2.52e+04
	Std	1.57e+03	3.48e+03	3.38e+03	4.19e+03	3.87e+03	2.32e+03	**1.24e+03**	2.73e+03

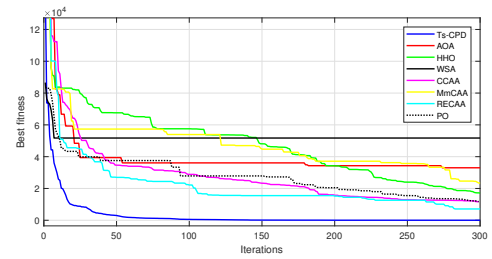
Table 2. Performance of metaheuristic algorithms compared with Ts-CPD on 30-dimensional multimodal problems.

Benchmark		Ts-CPD	AOA	HHO	WSA	CCAA	MmCAA	RECAA	PO
f_6	Avg	**2.40e+03**	1.10e+10	1.32e+08	1.00e+08	2.56e+09	4.27e+07	2.00e+06	2.98e+09
	Std	4.84e+03	3.16e+09	1.20e+08	**0.00e+00**	1.93e+09	1.67e+07	1.19e+06	1.98e+09
f_7	Avg	**4.72e+00**	1.32e+03	3.15e+02	3.61e+03	5.85e+01	1.40e+02	5.57e+01	4.81e+01
	Std	**7.72e+00**	1.68e+02	7.29e+01	5.20e+02	1.47e+01	3.44e+01	1.70e+01	2.70e+01
f_8	Avg	2.10e+01	2.11e+01	2.09e+01	2.11e+01	2.10e+01	2.10e+01	2.10e+01	**2.05e+01**
	Std	**5.48e-02**	8.26e-02	8.58e-02	5.72e-02	8.31e-02	6.76e-02	6.42e-02	7.97e-02
f_9	Avg	**1.02e+02**	3.04e+02	3.14e+02	4.01e+02	2.24e+02	2.31e+02	1.63e+02	2.50e+02
	Std	2.64e+01	2.18e+01	2.60e+01	**1.34e+01**	2.52e+01	1.70e+01	2.06e+01	4.29e+01
f_{10}	Avg	**4.89e+01**	4.66e+02	4.21e+02	7.25e+02	5.18e+02	3.93e+02	2.54e+02	4.05e+02
	Std	**1.45e+01**	4.68e+01	7.25e+01	5.27e+01	6.17e+01	5.68e+01	2.38e+01	4.22e+01
f_{11}	Avg	**2.16e+01**	4.17e+01	4.07e+01	4.37e+01	2.97e+01	3.55e+01	3.27e+01	3.04e+01
	Std	2.57e+00	2.49e+00	2.56e+00	2.12e+00	2.07e+00	2.01e+00	**1.41e+00**	8.22e+00
f_{12}	Avg	1.29e+06	1.50e+06	7.22e+05	1.50e+06	7.89e+05	**6.94e+05**	7.14e+05	9.86e+05
	Std	1.44e+05	2.46e+05	2.00e+05	1.66e+05	1.39e+05	1.67e+05	1.25e+05	**7.79e+03**
f_{13}	Avg	**6.45e+00**	2.82e+01	3.31e+01	1.27e+02	1.46e+01	2.20e+01	1.76e+01	1.15e+01
	Std	4.29e+00	5.82e+00	5.18e+00	3.60e+01	2.80e+00	**1.40e+00**	1.72e+00	3.09e+00
f_{14}	Avg	**1.30e+01**	1.35e+01	1.37e+01	1.39e+01	1.34e+01	1.36e+01	1.34e+01	1.41e+01
	Std	3.26e-01	3.42e-01	2.04e-01	2.21e-01	2.90e-01	**1.35e-01**	2.26e-01	1.43e-01
f_{15}	Avg	5.69e+02	9.80e+02	7.41e+02	1.24e+03	6.92e+02	5.69e+02	**5.13e+02**	1.03e+03
	Std	1.31e+02	7.26e+01	1.26e+02	8.68e+01	1.08e+02	**4.77e+01**	5.78e+01	1.16e+02
f_{16}	Avg	**2.91e+02**	8.22e+02	4.92e+02	1.17e+03	5.57e+02	4.04e+02	3.07e+02	6.98e+02
	Std	1.79e+02	9.05e+01	8.22e+01	1.41e+02	1.05e+02	5.04e+01	**4.74e+01**	1.03e+02
f_{17}	Avg	**2.90e+02**	8.79e+02	5.91e+02	1.18e+03	6.23e+02	4.83e+02	3.52e+02	7.86e+02
	Std	1.92e+02	1.40e+02	7.63e+01	1.61e+02	1.21e+02	6.79e+01	**5.10e+01**	9.67e+01
f_{18}	Avg	9.77e+02	9.77e+02	**9.00e+02**	9.00e+02	**9.00e+02**	**9.00e+02**	**9.00e+02**	**9.00e+02**
	Std	6.41e+01	1.44e+02	**0.00e+00**	3.95e-06	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**

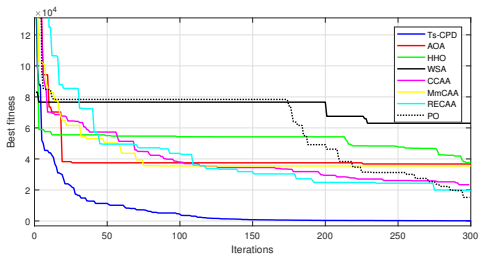
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

Table 2. *Cont.*

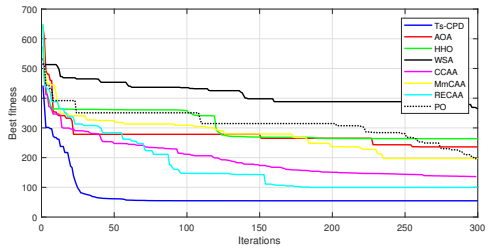
f_{19}	Avg	9.72e+02	1.01e+03	**9.00e+02**	9.00e+02	**9.00e+02**	**9.00e+02**	**9.00e+02**	**9.00e+02**
	Std	3.42e+01	1.53e+02	**0.00e+00**	4.51e-06	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f_{20}	Avg	9.79e+02	9.73e+02	**9.00e+02**	9.00e+02	**9.00e+02**	**9.00e+02**	**9.00e+02**	**9.00e+02**
	Std	3.99e+01	1.36e+02	**0.00e+00**	4.48e-06	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f_{21}	Avg	9.64e+02	1.31e+03	1.21e+03	1.40e+03	1.33e+03	1.25e+03	**8.36e+02**	1.14e+03
	Std	3.34e+02	1.57e+01	1.09e+02	**1.18e+01**	3.48e+01	7.07e+01	1.32e+02	1.95e+01
f_{22}	Avg	**9.21e+02**	1.42e+03	1.28e+03	1.82e+03	1.24e+03	1.13e+03	1.05e+03	1.09e+03
	Std	**1.66e+01**	7.23e+01	1.20e+02	1.05e+02	9.86e+01	3.48e+01	4.37e+01	8.81e+01
f_{23}	Avg	1.07e+03	1.31e+03	1.24e+03	1.40e+03	1.35e+03	1.25e+03	**9.26e+02**	1.15e+03
	Std	2.02e+02	1.69e+01	8.21e+01	**1.39e+01**	2.77e+01	4.86e+01	1.66e+02	2.41e+01
f_{24}	Avg	**2.26e+02**	1.37e+03	1.34e+03	1.46e+03	1.39e+03	1.33e+03	9.69e+02	1.07e+03
	Std	6.48e+01	1.82e+01	7.33e+01	**1.39e+01**	4.43e+01	4.04e+01	1.90e+02	1.67e+02
f_{25}	Avg	**1.01e+03**	1.38e+03	1.40e+03	1.47e+03	1.40e+03	1.39e+03	1.23e+03	1.28e+03
	Std	9.32e+00	2.66e+01	2.99e+01	**8.30e+00**	3.26e+01	1.90e+01	4.54e+01	1.26e+02



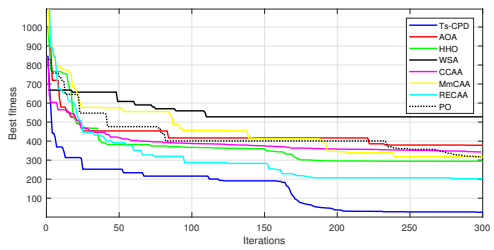
(a) f_2



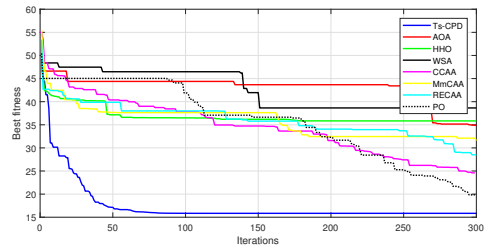
(b) f_4



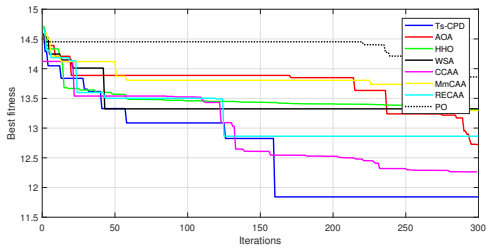
(c) f_9



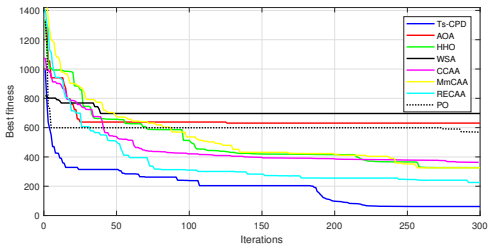
(d) f_{10}



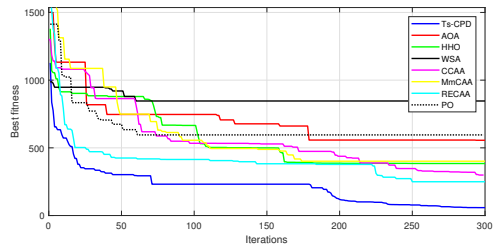
(e) f_{11}



(f) f_{14}



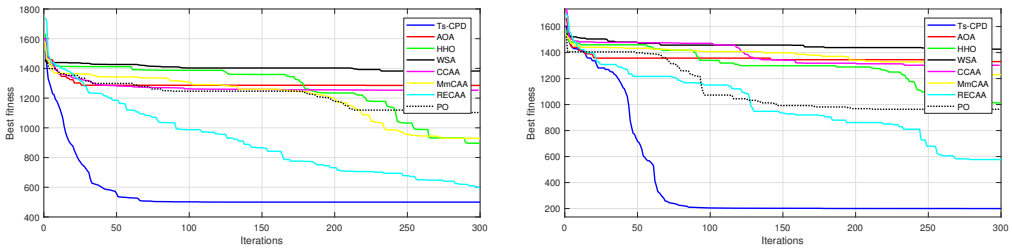
(g) f_{16}



(h) f_{17}

Figure 2. *Cont.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1



(i) f_{21} (j) f_{24}

Figure 2. Convergence curves of the different algorithms for CEC05 functions in 30 dimensions.

4. The proposed tool for analog IC sizing

The EDA tool proposed for the designer of analog circuits through the Ts-CPD algorithm allows obtaining a minimum area of the components used while complying with the design specifications. It is handy for designing the frequency response of circuits, such as bandwidth, phase margin, CMRR, or PSRR; only the slew rate can be designed in the time domain. For this purpose, before beginning the design, the designer must introduce the specifications (restrictions) of the circuit and the acceptable ranges and values for the parameters according to the technology used. The parameters to choose are the width and length of the CMOS transistors, capacitance and resistance (if any) values, bias current, and voltage sources.

The tool consists of two main modules: the optimization and synthesis processes. The optimization process contains the Ts-CPD algorithm comprising the CPSO-DE and the Deb rule, with a new transition rule given by (13); this module is implemented in Matlab. The synthesis process uses the specialized Ngspice software, which allows analog circuit simulations without mathematical equations. Instead, the standard configurations necessary to evaluate the performance of circuits are implemented in a netlist format. Both modules, the optimization and synthesis processes, are linked, allowing an automatic circuit design. The flow chart for our EDA tool, using Ts-CPD, is shown in Figure 3.

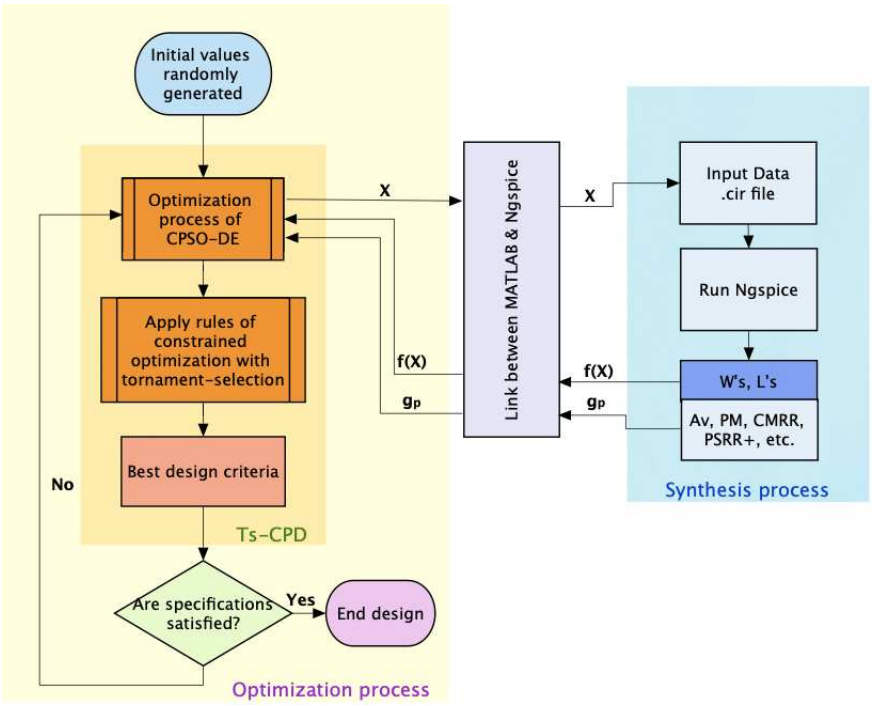


Figure 3. Flow chart of Ts-CPD as part of an EDA Tool.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

The following subsection describes three case studies, in terms of their variables and constraints, that will be used to verify the efficiency of the EDA tool.

4.1. Cases of study

To test our algorithm and tool, we chose three case studies, a “CMOS Differential Amplifier" differential amplifier, a “CMOS two-stage operational amplifier," and a “CMOS folded cascode operational transconductance amplifier." These cases were chosen because they have already been studied previously, and therefore, it is possible to compare the results of our algorithm against previous results, which is very interesting. In this sense, case 1 has 5 independent variables and 11 restrictions to meet, case 2 has 5 independent variables and 11 restrictions, while case 3, the most complete, has 9 independent variables and 13 restrictions to meet at the same time.

4.1.1. Case 1: CMOS differential amplifier

Figure 4 shows our first case of study, a CMOS differential amplifier. First, M_1 must be equally sized than M_2; thus, the following equality restrictions must be satisfied:

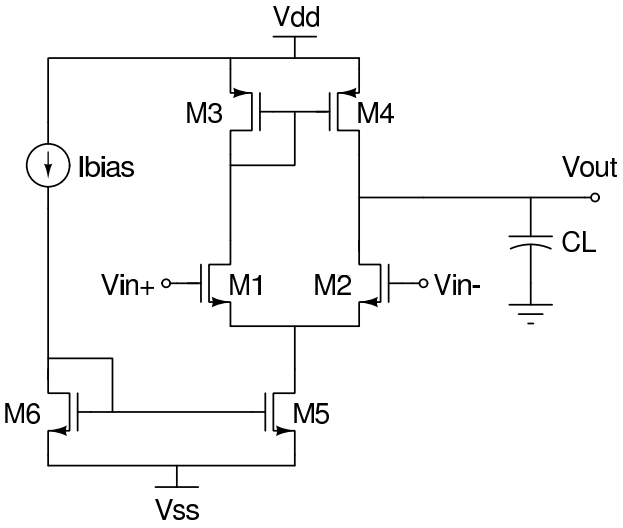


Figure 4. CMOS differential amplifier.

$$W_1 = W_2 \quad \text{and} \quad L_1 = L_2. \tag{14}$$

Secondly, s of the current source, M_3 and M_4, must be equally sized, too, thus

$$W_3 = W_4 \quad \text{and} \quad L_3 = L_4. \tag{15}$$

We let both W_5 and W_6 be independent variables, and our algorithm selects their values while $L_5 = L_6$. That is because the sizes of all s are within a specific range imposed by the technology used for this design:

$$W_{n,min} \leq W_n < W_{n,max} \quad , \quad n = 1, 2, \cdots, 6. \tag{16}$$

In our case, $W_{n,min}$ was fixed to 4μm for a better comparison with other works, and $W_{n,max}$ was fixed to 120μm to have a value large enough. For this example, there are 5 independent variables (W_1, W_3, W_5, W_6 and I_{bias}) and 2 dependent ones (W_2 and W_4). On the other hand, the design specifications to be met will be treated as constraints. For this case, there are 11 constraints: load capacitance, slew rate, power dissipation, phase margin, cut-off frequency, DC gain, V_{IC}(min), V_{IC}(max), CMRR, PSRR+ and PSRR-.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

4.1.2. Case 2: CMOS two-stage operational amplifier

Figure 5 shows our second case of study, a CMOS two-stage operational amplifier consisting of 8 s. The first amplification stage, differential input, has the stipulation that M_1 must be equally sized as M_2, so that equations (14), (15) are still valid, and we add,

$$W_5 = W_8 \quad \text{and} \quad L_5 = L_8. \tag{17}$$

Also, to avoid an output offset at the second amplification stage, the following restriction is imposed:

$$\frac{W_7/L_7}{W_5/L_5} = 2\frac{W_6/L_6}{W_4/L_4}. \tag{18}$$

Similarly, as in (16), sizes of the CMOS two-stage operational amplifier are in a specific range, but now $n = 8$. Also, the compensation capacitance is within a range of values, between $C_{C,min}$ and $C_{C,max}$, which the designer selects:

$$C_{C,min} \leq C_C < C_{C,max}. \tag{19}$$

The $C_{C,min}$ and $C_{C,max}$ values are fed to the Ts-CPD algorithm through a file in our EDA tool. We choose $C_{C,min} = 2$pF, because lower values than that are challenging to achieve and $C_{C,max} = 14$pF to avoid using significant areas, but these values are easily changed.

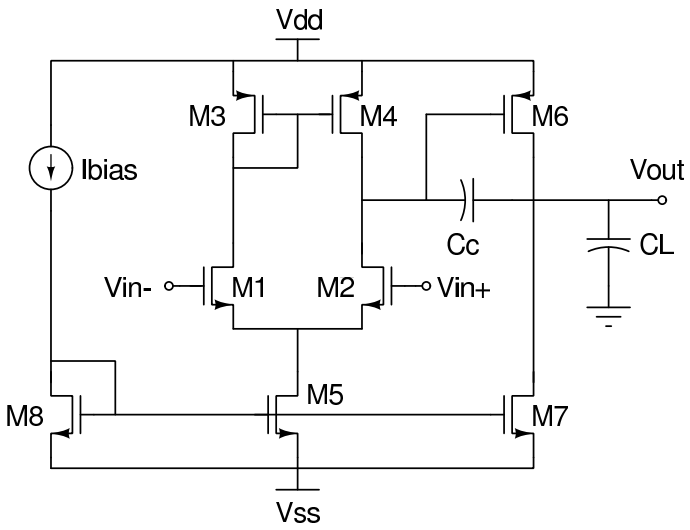


Figure 5. CMOS two-stage operational amplifier

On the other hand, bias current I_{BIAS} also is within a range o values:

$$I_{BIAS,min} \leq I_{BIAS} < I_{BIAS,max}. \tag{20}$$

It is clear from equations (14), (15) and (17) that, for the purpose of design, W_2, W_4 and W_6 can be handled as independent variables, while W_1, W_3 and W_5 as can be handled as dependent ones. W_7 is deduced from (18), thus, W_7 is also a dependent variable; I_{BIAS} and C_C are considered independent variables whose values are bounded by (20) and (19), respectively. Therefore, this example has 5 independent variables, W_2, W_4, W_6, I_{BIAS} and C_c, whose values are selected by our algorithm and 5 dependent variables W_1, W_3, W_5, W_7 and W_8, whose impact over cost function and restrictions is evaluated by our algorithm to determine new values for independent variables, in an iterative process. In this paper, the length of s is considered constant. However, when lengths are considered variables, the minimum and maximum values must be established, as for widths in equation (16). For this case,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

there are 11 constraints: load capacitance, slew rate, power dissipation, phase margin, unity gain bandwidth, DC gain, V_{IC}(min), V_{IC}(max), CMRR, PSRR+, and PSRR-.

4.1.3. Case 3: CMOS folded cascode operational transconductance amplifier

A third case of study is the folded cascode operational transconductance amplifier (FCOTA) shown in Figure 6. The transistors M_1 and M_2 are equally sized; thus, equation (14) is also valid. We considered the transistor widths W_3 and W_4 independent variables and W_5 and W_{14} dependent ones, as follows:

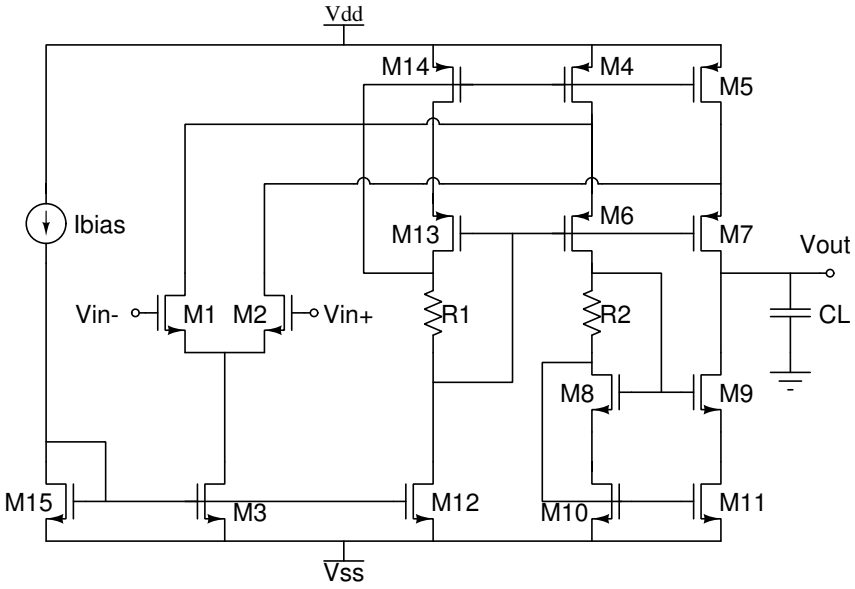


Figure 6. CMOS folded cascode operational transconductance amplifier

$$W_4 = W_5 = W_{14} \quad \text{and} \quad L_4 = L_5 = L_{14}. \tag{21}$$

In addition, W_6, W_8, W_{12} and W_{15} are considered independent variables while W_7, W_9, W_{10}, W_{11}, and W_{13} are considered dependent variables, as follows:

$$W_6 = W_7 = W_{13} \quad \text{and} \quad L_6 = L_7 = L_{13}, \tag{22}$$

$$W_8 = W_9 = W_{10} = W_{11} \quad \text{and} \quad L_8 = L_9 = L_{10} = L_{11}. \tag{23}$$

Table 3. Design criteria for CMOS differential amplifier (Case 1) and results obtained with several evolutionary algorithms.

Design criteria	Specs.	Ts-CPD	MOL[59]	SOA[60]	PSO[19]	HS[44]	DE[44]	ABC[44]	GA[43]
Load capacitance (pF)	≥ 2	2.1	5	3.5	5	5	5	5	2
Slew rate (V/μs)	$\geq$10	24.3	10	12.28	22.4	14.916	18.451	15.67	3.2
Power dissipation (μW)	$\leq$2,000	1,075	863	117	1,260	886	990	830	31
Phase margin (°)	> 45	86.1	89	83.73	83.8	89.1	88.81	91.248	72
Cut-off frequency (KHz)	$\geq$100	100.5	-	104.8	100	114	129.7	112.367	-
Unity gain bandwidth (MHz)	≥ 1	10	17.87	12.5	12.3	-	-	-	3.8
DC gain (dB)	$\geq$40	40.3	30	44.02	42	40.98	41.23	42.045	60
$V_{IC(min)}$ (V)	$\geq$-1.5	-0.8	-0.5	-0.37	-0.8	-0.7	-0.92	-0.97	-1.3
$V_{IC(max)}$ (V)	$\leq$2	1.1	0.7	1.57	1.4	1.2	1.15	1.2	1.9
CMRR (dB)	$>$40	81.0	59	83.17	84.2	78.5	78.39	79.67	-
PSRR$^+$ (dB)	$>$40	41.2	41	60.59	40.1	42.93	43.14	43.857	-
PSRR$^-$ (dB)	$>$40	78.1	68	108.6	68	67.64	68.175	68.423	-
Total component area (μm^2)	$<$300	**109**	235	236	296	-	-	-	6,500
AFOM$_{SS}$ (MHz·pF)/(μW·mm^2)		179	**457**	318	165	-	-	-	40

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

The values of the bias current I_{BIAS} are bounded by (20) and properly selected by our algorithm. For design, we considered R_1 as an independent variable. Thus, our algorithm also selects its value within $1000 < R_1 < 6000$, while R_2 is considered a dependent variable, with $R_1 = R_2$. This way, there are 9 independent variables (W_1, W_3, W_4, W_6, W_8, W_{12}, W_{15}, I_{BIAS} and R_1) and 9 dependent variables (W_2, W_5, W_7, W_9, W_{10}, W_{11}, W_{13}, W_{14} and R_2). The constraints for this case are 13: load capacitance, slew rate, power dissipation, phase margin, unity gain bandwidth, DC gain, V_{IC}(min), V_{IC}(max), V_{out}(min), V_{out}(max), CMRR, PSRR+ and PSRR-.

5. Numerical results and discussion

In order to test our proposed tool, three examples of design are shown in this section. First, the optimization is implemented in MATLAB R2014b, while the simulation of circuits is implemented in the NGSPICE r26 simulator; both are linked, so the design process is completely automated. On the other hand, the model of NMOS and PMOS transistors for 0.35 μm technology was downloaded from the MOSIS database. Finally, the transistor lengths were set to fixed values close to those in the literature for comparison purposes.

Our design objective is to minimize the area of analog circuits. However, designing an amplifier is always a trade-off, so we introduce the small-signal figure of merit that considers silicon area to assess the designed circuits' overall performance [61]:

$$AFOM_{SS} = (f_u \cdot C_L / P^Q \cdot Area) \tag{24}$$

where f_u is the unity gain frequency, C_L is the load capacitance, P^Q is the power consumption at quiescent, and Area is the component (transistors) area.

5.1. Numerical results for CMOS differential amplifier (Case 1)

As a first example, the differential amplifier of Figure 4 is designed. We aim to minimize the total component area, which is our cost function, below $300\mu m^2$ while restrictions are still met. As shown in Table 3, the power dissipation is specified to be $< 2,200\mu$W, DC gain $\geq$40 dB, slew rate $\geq$ 10 V/μs and the cut-off frequency $\geq$ 100KHz. Other specifications are Common Mode Rejection Ratio (CMRR), Positive Power Supply Rejection Ratio (PSRR+), Negative Power Supply Rejection Ratio (PSRR-), and the Input Common-Mode Range (ICMR), all to be $>$ 40dB, and finally $V_{IC(min)} \geq -1.5$V and $V_{IC(max)} \leq$ 2V. The circuit's load determines load capacitance, but the specification to be satisfied is $\geq$ 2pF; we choose 2.1 pF. The AFOM$_{SS}$ is also shown.

For the optimization purpose, some variables are set to a fixed value, and the micro-channel lengths were set to $L_1 = L_2 = L_3 = L_4 = 3.5\mu$m, $L_5 = L_6 = 1.4\mu$m, and voltage sources were set to $V_{dd} = -V_{ss} = 2.5$V. On the other hand, C_c and I_{bias} are treated as independent variables with restrictions , *i.e.* they can run within a specific range of values in our algorithm.

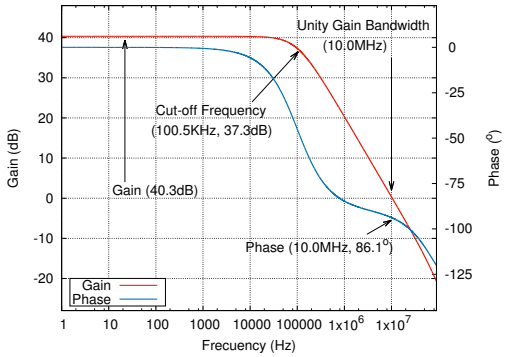
The numerical results for the differential amplifier of Figure 4 are shown in Table 3; it presents a comparison of Ts-CPD with several methods: MOL [59], SOA [60], PSO [19], HS [44], DE [44], Artificial Bee Colony (ABC) [44], and GA [43]. The Ts-CPD obtains the lower total component area for methods that report this design objective and obtains the higher slew rate and PSRR$^-$; other specifications are also accomplished. Here, the MOL algorithm has the higher AFOM$_{SS}$ value. Table 4 shows the result of the designed differential amplifier for three evolutionary algorithms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

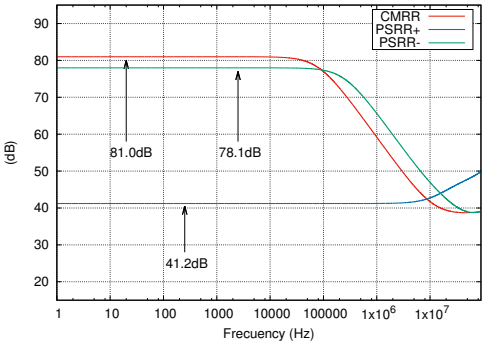
Table 4. Design parameters for three algorithms (Case 1).

Design parameters	Ts-CPD	PSO[19]	GA[43]
W_1/L_1 (μm/μm)	7.6/3.5	29.4/3.5	240/13.2
W_2/L_2 (μm/μm)	7.6/3.5	29.4/3.5	240/13.2
W_3/L_3 (μm/μm)	4.6/3.5	11.3/3.5	7.3/7.7
W_4/L_4 (μm/μm)	4.6/3.5	11.3/3.5	7.3/7.7
W_5/L_5 (μm/μm)	5.9/1.4	4.2/1.4	4.6/2.4
W_6/L_6 (μm/μm)	11.2/1.4	4.2/1.4	2.4/2.4
I_{bias} (μA)	141	125	2

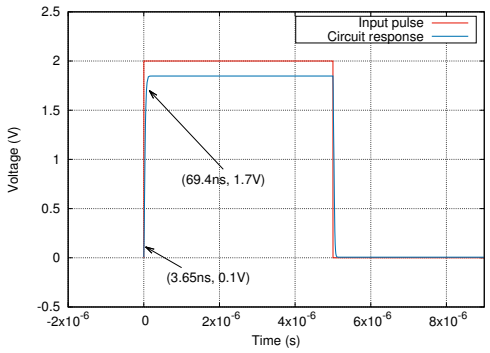
In order to explore the performance of the differential amplifier designed, we show the DC gain and phase margin in Figure 7a; The CMRR, PSRR+, and PSRR- in Figure 7b; Slew rate in Figure 7c; and the ICMR in Figure 7d, which is used for the graphical determination of $V_{IC(min)}$ and $V_{IC(max)}$. These graphics demonstrate that the designed circuit behaves well and is accomplished with all the constraints (Specifications).



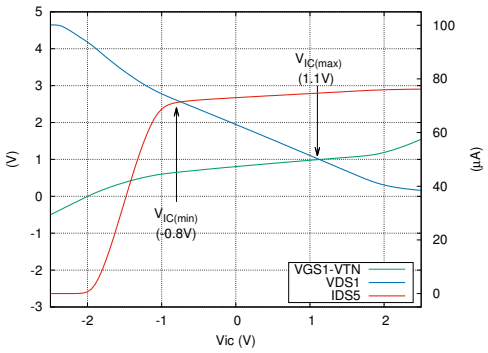
(a) Gain and phase.



(b) CMRR, PSRR+ and PSRR- .



(c) Slew rate.

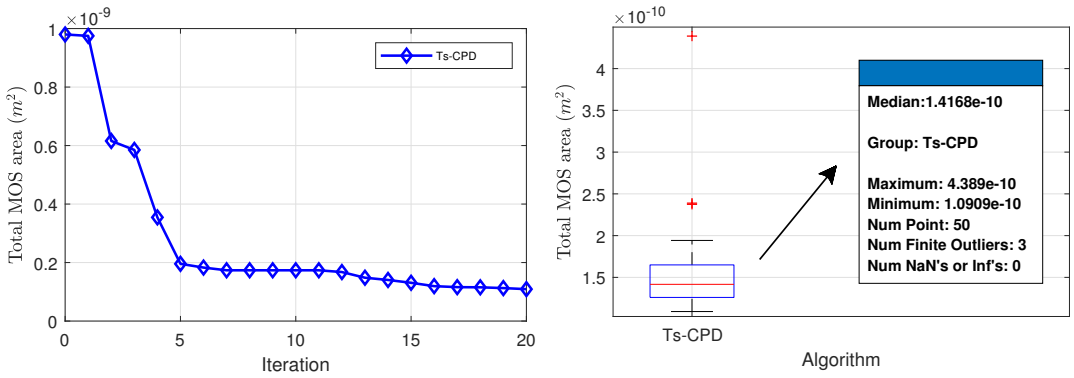


(d) ICMR.

Figure 7. Performance of CMOS differential amplifier.

Figure 8a shows the convergence of our algorithm for this circuit design, which has an excellent profile. Our algorithm's behavior was also tested with 50 runs; the corresponding Box and Whisker plot is shown in Figure 8b. The median is 1.4168x10^{-10}m^2, which is still below the results reported for other algorithms; see Table 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1



(a) Ts-CPD convergence profile. **(b)** Ts-CPD Box and Whisker plot.

Figure 8. Ts-CPD test for CMOS differential amplifier.

5.2. Numerical results for CMOS two-stage operational amplifier (Case 2)

As a second example, we designed the two-stage operational amplifier in Figure 5. Again, the aim is to minimize the total component area as much as possible while constraints are still met. The total component area is specified to be $< 300\mu m^2$, and in this case, the DC gain $>$ 60dB, unity gain bandwidth $\geq$ 3MHz, phase margin $\geq 45^o$, slew rate $\geq$ 10 V/μs and load capacitance $\geq$ 7pF. In other set of specifications, CMRR$>$ 60dB, PSRR$^+$ $>$ 70dB, PSRR$^+$ $>$ 70dB, $V_{IC(min)} > -1.5$V and $V_{IC(max)} \leq$ 2V. At the end, the AFOM$_{SS}$ is shown.

The microchannel lengths of all MOS transistors have been set to a fixed value, $L_1 = L_2 \cdots L_8 = 0.8\mu$m, while voltage sources are set to $V_{dd} = -V_{ss} = 2.5$V. Here, C_c and I_{bias} are independent variables. Thus, our algorithm determines its values in concordance with (19) and (20), respectively.

Table 5 shows the complete set of restrictions and design objective for the CMOS operational amplifier of Figure 5, as well as the comparison of methods Ts-CPD, GSA-PSO[20], PSO, and Geometric Programming (GP) [9]. As expected, the Ts-CPD has the lower component area and the highest slew rate and PSRR-. The AFOM$_{SS}$, on the other hand, is higher for our algorithm. The design parameters of the optimized circuit are shown in Table 6.

Table 5. Design criteria for CMOS two-stage operational amplifier and results obtained with several algorithms.

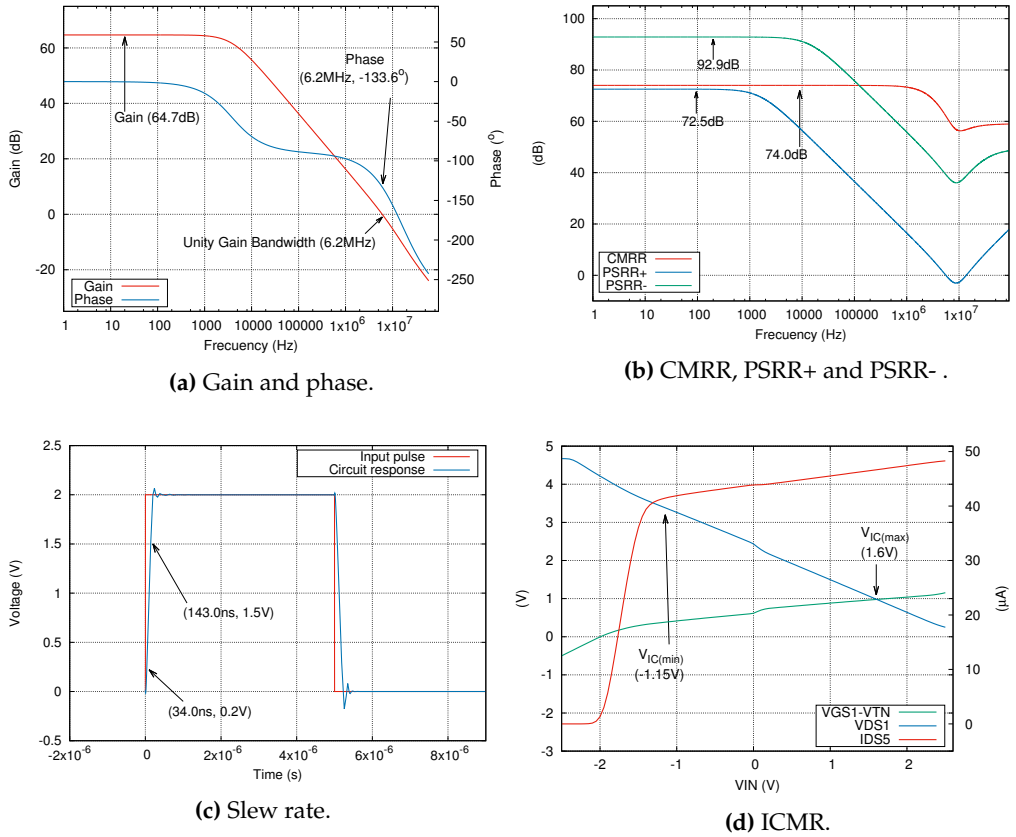
Design criteria	Specs.	Ts-CPD	GSA-PSO[20]	PSO[19]	GP[9]
Load capacitance (pF)	≥ 7	7.1	7.2	10	3
Slew rate (V/μs)	≥ 10	11.9	10.88	11.13	88
Power dissipation (μW)	$\leq$ 2,500	1,084	712.8	2,370	5,000
Phase margin (°)	> 45	46	66.2	66.55	60
Unity gain bandwidth (MHz)	$\geq$3	6.2	5.776	5.32	86
DC gain (dB)	$>$60	64.7	75.43	63.8	89.2
$V_{IC(min)}$ (V)	$\geq$-1.5	-1.15	-0.886	-0.8	-
$V_{IC(max)}$ (V)	$\leq$2	1.6	1.9	1.75	-
CMRR (dB)	$>$60	74.0	75.43	63.8	89.2
PSRR$^+$ (dB)	$>$70	72.5	83.2	78.27	116
PSRR$^-$ (dB)	$>$70	92.9	110.4	93.56	98.4
Total component area (μm^2)	$<$300	**45.6**	109.6	265	8,200
AFOM$_{SS}$ (MHz·pF)/(μW·mm^2)		**902**	532	85	6

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

Table 6. Design parameters for the four algorithms (Case 2).

Design variables	Ts-CPD	GSA-PSO[20]	PSO[19]	GP[9]
W_1/L_1 (μm/μm)	4.1/0.8	4/2	4.9/2	232.8/0.8
W_2/L_2 (μm/μm)	4.1/0.8	4/2	4.9/2	232.8/0.8
W_3/L_3 (μm/μm)	4.0/0.8	4/2	5.9/2	143.6/0.8
W_4/L_4 (μm/μm)	4.0/0.8	4/2	5.9/2	143.6/0.8
W_5/L_5 (μm/μm)	4.7/0.8	2.8/2	2.1/2	64.6/0.8
W_6/L_6 (μm/μm)	19.8/0.8	24/2	90.9/2	588.8/0.8
W_7/L_7 (μm/μm)	11.5/0.8	9.2/2	16.3/2	132.6/0.8
W_8/L_8 (μm/μm)	4.7/0.8	2.8/2	2.1/2	2/0.8
C_C (pF)	3.8	2.8	3	3.5
I_{bias} (μA)	42.7	28	40.39	10

The performance of the CMOS two-stage operational amplifier can be evaluated through the gain and phase plot in Figure 9a; the CMRR, PSRR+, and PSRR- plots in Figure 9b; the ICMR in Figure 9c; and the slew rate in Figure 9d. These plots also demonstrate the excellent performance of the designed circuit.



(a) Gain and phase.

(b) CMRR, PSRR+ and PSRR- .

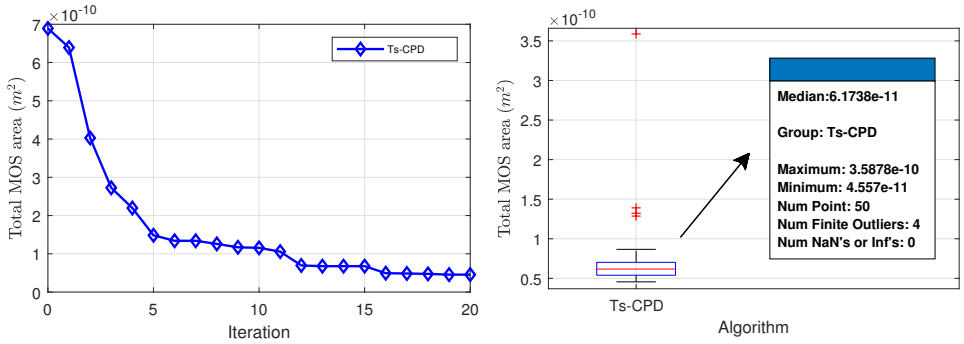
(c) Slew rate.

(d) ICMR.

Figure 9. Performance of CMOS two-stage operational amplifier.

On the other hand, we evaluated the performance of our algorithm with the convergence profile in Figure 10a, and the Box and Whisker plot of Figure 10b. After 16 iterations, the Ts-CPD reached convergence; see Figure 10a. We executed 50 trial runs for the circuit design; Figure 10b shows the corresponding Box and Whisker plot for the total MOS area of transistors. The best value is $4.557\text{x}10^{-11}\text{m}^2$, but the median ($6.1738\text{x}10^{-11}\text{m}^2$) is also lower than others reported for this circuit, as can be seen in Table 5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1



(a) Ts-CPD convergence profile. **(b)** Ts-CPD Box and Whisker plot.

Figure 10. Ts-CPD test for CMOS two-stage operational amplifier.

5.3. Numerical results for CMOS folded cascode operational transconductance amplifier (Case 3)

Our third example is the folded cascode operational amplifier shown in Figure 6. The total component area specified is $< 1315.9\ \mu\text{m}^2$ (our design objective). At the same time, specified constraints are gain $>$ 74dB, unity bandwidth $\geq$ 10MHz, phase margin $> 60^o$, slew rate $\geq 10V/\mu s$ and load capacitance $\geq$ 10pF (we chose exactly 10.0pF). More constraints are CMRR, PSRR+, PSRR- all three $\geq$ 55dB, $V_{IC(min)} \geq -1.5$, $V_{IC(max)} \leq 2.5$, and finally $V_{out(min)} \geq -2$ and $V_{out(max)} \leq 2$. And at the end, the AFOM$_{SS}$ is shown.

For all MOS transistors, the lengths have been set to a fixed value, $L_1 = L_2 \cdots L_{15} = 1.5\mu$m, and the voltage sources are set to $V_{dd} = -V_{ss} = 2.5$V. Besides the transistor widths (W_i), I_{bias}, R_1 and R_2 are also variables.

Table 7 shows the numerical results for the FCOTA of Figure 6 and a comparison of methods Ts-CPD and ALC-PSO[1]. Our proposal, Ts-CPD, has the lower total component area (our design objective) and the highest Unity gain bandwidth, phase margin, CMRR, and PSRR-, while other constraints are also met. Additionally, the AFOM$_{SS}$ is greater for our algorithm. The parameters of the optimized circuit for the two proposals are shown in Table 8.

Table 7. Design criteria for CMOS folded cascode operational transconductance amplifier.

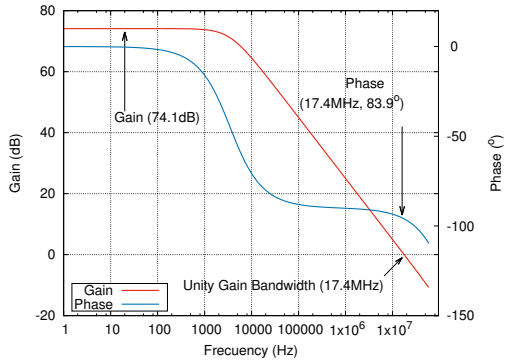
Design criteria	Specs.	Ts-CPD	ALC-PSO[1]
Load capacitance (pF)	≥ 10	10.0	10.028
Slew rate (V/μs)	≥ 10	13.8	19.37
Power dissipation (mW)	≤ 5	3.3	2.504
Phase margin ($^\circ$)	> 60	83.9	63.1
Unity gain bandwidth (MHz)	$\geq$10	17.8	11.11
DC gain (dB)	$>$74	74.1	76.97
$V_{IC(min)}$ (V)	$\geq$-1.5	-0.69	-1.466
$V_{IC(max)}$ (V)	$\leq$2.5	2.41	2.486
$V_{out(min)}$ (V)	$\geq$-2	-2.0	-1.936
$V_{out(max)}$ (V)	$\leq$2	1.99	1.996
CMRR (dB)	$>$55	111.8	87.58
PSRR$^+$ (dB)	$>$55	82.9	84.21
PSRR$^-$ (dB)	$>$55	74.6	61.47
Total component area (μm^2)	$<$1315.9	**600.9**	835.2625
AFOM$_{SS}$ (MHz·pF)/(μW·mm^2)		**89,764**	53,269

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

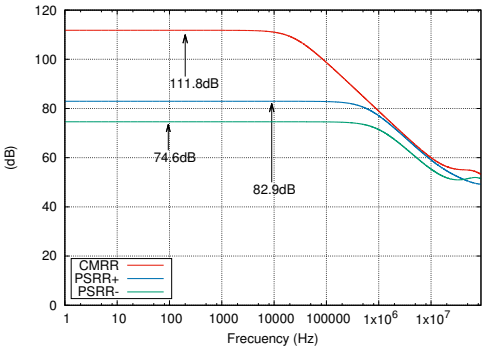
Table 8. Design parameters for Case 3.

Design variables	Ts-CPD	ALC-PSO[1]
W_1/L_1 (μm/μm)	48.43/1.25	60.46/1.25
W_2/L_2 (μm/μm)	48.43/1.25	60.46/1.25
W_3/L_3 (μm/μm)	78.66/1.25	35.8/1.25
W_4/L_4 (μm/μm)	13.40/1.25	40.1/1.25
W_5/L_5 (μm/μm)	13.40/1.25	40.1/1.25
W_6/L_6 (μm/μm)	24.26/1.25	45.94/1.25
W_7/L_7 (μm/μm)	24.26/1.25	45.1/1.25
W_8/L_8 (μm/μm)	25.35/1.25	59.63/1.25
W_9/L_9 (μm/μm)	25.35/1.25	59.63/1.25
W_{10}/L_{10} (μm/μm)	25.35/1.25	59.63/1.25
W_{11}/L_{11} (μm/μm)	25.35/1.25	59.63/1.25
W_{12}/L_{12} (μm/μm)	55.60/1.25	14.85/1.25
W_{13}/L_{13} (μm/μm)	24.26/1.25	45.94/1.25
W_{14}/L_{14} (μm/μm)	13.34/1.25	40.1/1.25
W_{15}/L_{15} (μm/μm)	35.23/1.25	-
I_{bias} (μA)	119.3	-
$R_1(k\Omega)$	4.83	1.89
$R_2(k\Omega)$	4.83	1.89

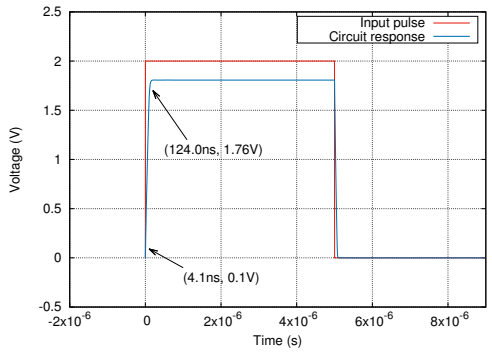
The excellent performance of the CMOS folded cascode operational transconductance amplifier is demonstrated through the plots of gain and phase in Figure 11a; CMRR, PSRR+, and PSRR- in Figure 11b; the ICMR in Figure 11c; and the slew rate, Figure 11d.



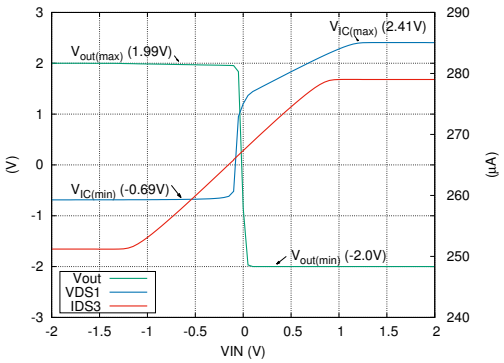
(a) Gain and phase.



(b) CMRR, PSRR+ and PSRR- .



(c) Slew rate.



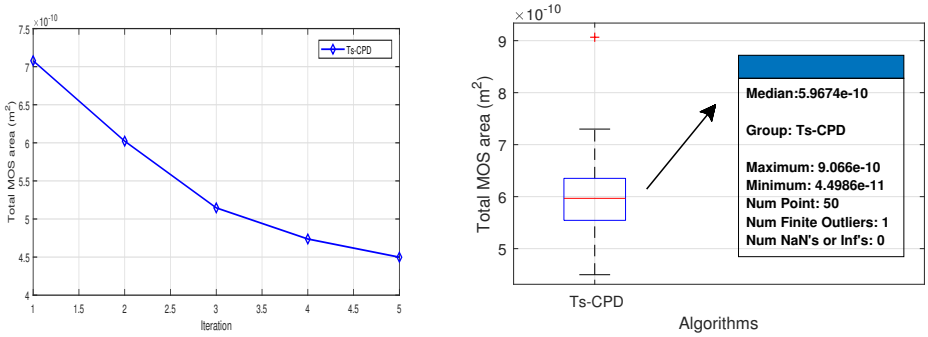
(d) ICMR.

Figure 11. Performance of CMOS folded cascode operational transconductance amplifier.

The Ts-CPD performance is evaluated with the convergence profile shown in Figure 12a and the Box and Whisker plot of Figure 12b. As can be seen in Figure 12a, the Ts-CPD converges very

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 October 2023 doi:10.20944/preprints202310.0597.v1

quickly for this circuit design in just 5 iterations. Figure 12b shows the Box and Whisker plot for 50 trial runs for the total MOS area of transistors. The median is 5.9674x10^{-11}m^2, and the solutions are very clustered towards this value.



(a) Ts-CPD convergence profile. **(b)** Ts-CPD Box and Whisker plot.

Figure 12. Ts-CPD test for CMOS folded cascode operational transconductance amplifier.

6. Conclusions

The Ts-CPSO algorithm that was proposed and implemented improves the CPSO by incorporating a way of evaluating the performance of constraints, through the optimization-with-constraints method, with a new rule we proposed. This algorithm has the advantage of not only minimizing the objective function but also ensuring that the constraints are met and then generating the new parameter values. Then the Ts-CPSO algorithm is incorporated into our EDA tool for the optimal sizing of analog circuits, which does not require mathematical equations since the optimization is linked to a simulator that provides the circuit's behavior.

The Ts-CPD algorithm, as part of our EDA tool, was tested with three cases of study in a 0.35um CMOS technology, a differential amplifier, a two-stage operational amplifier, and a folded cascode operational transconductance amplifier. It was proposed as a design objective to reduce the total area occupied by the transistors while complying with some established constraints. In all cases, our tool found a better solution, for the objective, than previously reported tools, while the constraints were kept within the desired limits.

In future work, we are going to implement a multi-objective algorithm, which we will add as the kernel of our EDA tool. We will also do design tests with analog circuits with more transistors and large-scale analog circuits, such as the ADC, considering the Layout design. As another potential future project, a framework incorporating multiple algorithms for optimizing various analog circuits can be developed. This framework would allow users to customize each algorithm's parameters to enhance its performance, compare the different methods with convergence plots and identify the optimal design.

Author Contributions: Conceptualization, P. L.-E. and P.M.-R.; methodology, P. L.-E. and P.M.-R.; validation, J.C.S.-T.-M. and N.H.-R.; formal analysis, P. L.-E., P.M.-R. and J.C.S.-T.-M.; investigation, P. L.-E., P.M.-R., J.C.S.-T.-M. and N.H.-R.; resources, P.M.-M. and J.C.S.-T.-M.; writing—original draft preparation, P. L.-E., P.M.-R., J.C.S.-T.-M. and N.H.-R.; writing—review and editing, P.M.-R. and J.C.S.-T.-M. visualization, P. L.-E., P.M.-R. and J.C.S.-T.-M.; supervision, J.C.S.-T.-M. and N.H.-R.; funding acquisition, J.C.S.-T.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Autonomous University of Hidalgo (UAEH) and the National Council for Humanities, Science and Technology (CONAHCYT) with project number F003-320109.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations were used in this research:

CPSO-DE	Cellular particle swarm algorithm with differential evolution
EDA	Electronic design automation
VLSI	Very Large Scale of Integration
CAD	Computer-Aided Design
Ts-CDP	Tournament-selection CPSO-DE

References

1. De, B.P.; Kar, R.; Mandal, D.; Ghoshal, S.P. An efficient design of CMOS comparator and folded cascode op-amp circuits using particle swarm optimization with an aging leader and challengers algorithm. *International Journal of Machine Learning and Cybernetics* **2016**, *7*, 325–344.
2. Dehbashian, M.; Maymandi-Nejad, M. An enhanced optimization kernel for analog IC design automation using the shrinking circles technique. *Engineering Applications of Artificial Intelligence* **2017**, *58*, 62–78.
3. Cohen, M.W.; Aga, M.; Weinberg, T. Genetic algorithm software for analog circuit design. *Procedia CIRP* **2015**, pp. 17–22.
4. Ozenli, D.; Alaybeyoglu, E.; Kuntman, H.; Cicekoglu, O. MOSFET-Only filter design automation based on polynomial regression with exemplary circuits. *AEU - International Journal of Electronics and Communications* **2018**, *84*, 342–354.
5. Durmuş, B. Optimal Components Selection for Active Filter Design with Average Differential Evolution Algorithm. *AEU - International Journal of Electronics and Communications* **2018**, *94*, 293–302.
6. Lourenço, N.; Martins, R.; Horta, N., Automatic Analog IC Sizing and Optimization Constrained with PVT Corners and Layout Effects; Springer International Publishing: Cham, 2017; chapter Previous Works on Automatic Analog IC Sizing, pp. 13–37. doi:10.1007/978-3-319-42037-0_2.
7. Brayton, R.K.; Hachtel, G.D.; Sangiovani-Vicentelli, A.L. A survey of optimization techniques for integrated-circuit design. Proceeding of the IEEE, 1981, Vol. 69, pp. 1334–1362.
8. Carley, L.R.; Rutenabar, R. How to automate analog IC designs. *IEEE Spectrum* **1988**, *25*, 26–30.
9. d. Hershenson, M.; Boyd, S.P.; Lee, T.H. Optimal Design of a CMOS op-amp via Geometric Programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2001**, *20*, 1–21.
10. d. Hershenson, M. Analog Circuit Design via Geometric Programming. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **2004**, *E87-A*, 298–310.
11. Fares, M.; Kaminska, B. FPAD: A Fuzzy Nonlinear Programming Aproach to Analog Circuit Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **1995**, *14*, 785–793.
12. Torralba, A.; Chávez, J.; Franquelo, L.G. Circuit Performance Modeling by Means of Fuzzy Logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **1996**, *15*, 1391–1398.
13. Oltean, G.; Miron, C.; Mocean, E. Multiobjective Optimization Method for analog circuits design based on fuzzy logic. 9th International Conference on Electronics, Circuits and Systems, 2002, Vol. 2, pp. 770–780.
14. Torralba, A.; Chávez, J.; Franquelo, L.G. FASY: A fuzzy-Logic Based Tool for Analog Synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **1996**, *15*, 705–715.
15. Miranda-Romagnoli, P.; Hernandez-Romero, N.; Seck-Tuoh-Mora, J.C. A neuro fuzzy solution in the design of analog circuits. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **2011**, *94*, 434–439.
16. de Lima Moreto, R.A.; Thomaz, C.E.; Gimenez, S.P. A customized genetic algorithm with in-loop robustness analyses to boost the optimization process of analog CMOS ICs. *Microelectronics Journal* **2019**, *92*, 104595.
17. Vural, R.A.; Yildirim, T. Component Value Selection for Analog Active Filter Using Particle Swarm Optimization. 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp. 25–28.

18. Mostafa, S.S.; Horta, N.; Ravelo-García, A.G.; Morgado-Dias, F. Analog Active Filter Design using a Multi Objective Genetic Algorithm. *AEU-International Journal of Electronics and Communications* **2018**, *93*, 83–94.
19. Vural, R.A.; Yildirim, T. Analog circuit sizing via swarm intelligence. *AEU - International Journal of Electronics and Communications* **2012**, *66*, 732–740.
20. Mallick, S.; Kar, R.; Mandal, D.; Ghoshal, S. Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization. *International Journal of Machine Learning and Cybernetics* **2017**, *8*, 309–331.
21. Ghosh, S.; De, B.P.; Kar, R.; Mandal, D.; Mal, A.K. Optimal design of complementary metal-oxide-semiconductor analogue circuits: An evolutionary approach. *Computers & Electrical Engineering* **2019**, *80*, 106485.
22. Sabat, S.L.; Kumar, K.S.; Udgata, S.K. Differential Evolution and Swarm Intelligence techniques for analog Circuit Synthesis. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC2009), 2009, pp. 469–473.
23. Barros, M.; Guilherme, J.; Horta, N. Analog circuits optimization based on evolutionary computation techniques. *Integration* **2010**, *43*, 136–155.
24. Jafari, A.; Bijami, E.; Rekabi, B.; Saeed, S. A design automation system for CMOS analog integrated circuits using New Hybrid Shuffled Frog Leaping Algoritm. *Microelectronics Journal* **2012**, *43*, 908–915.
25. Bausser, M.; Talbi, E.G.; Nebro, A.; Alba, E. Metaheuristics for multiobjective combinatorial optimization problems: review and recent issues. Research report, INRIA, 2006.
26. Chan, F.T.; Tiwari, M.K. *Swarm Intelligence*; IntechOpen: Rijeka, 2007. doi:10.5772/48.
27. Liu, B.; Gielen, G.; Fernández, F.V., Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach; Springer Berlin Heidelberg, 2014; chapter Basic Concepts and Background, pp. 1–17. doi:10.1007/978-3-642-39162-0_1.
28. Kumar, P.P.; Duraiswamy, K. An Optimized Device Sizing of Analog Circuits using Particle Swarm Optimization. *Journal of Computer Science* **2012**, *8*, 930–935.
29. Gupta, H.; Ghosh, B. Analog Circuits Design Using Ant Colony Optimization. *International Journal of Electronics, Computer and Communications Technologies* **2012**, *2*, 9–21.
30. Cheng, Y.; Chan, L.Y.; Chen, Y.L.; Liao, Y.C.; Liu, C.N. A bias-driven approach to improve the efficiency of automatic design optimization for CMOS OP-Amps. 2012 4th Asia Symposium on Quality Electronic Design (ASQED), 2012, pp. 59–63.
31. Tlelo-Cuautle, E.; Sanabria-Borbon, A.C. Optimising operational amplifiers by evolutionary algorithms and gm/Id method. *International Journal of Electronics* **2016**, *103*, 1665–1684.
32. Lourenço, N.; Martins, R.; Horta, N. *Automatic analog IC sizing and optimization constrained with PVT corners and layout effects*; Springer, 2017.
33. Gielen, G.G.E.; Maricau, E.; De Wit, P. Designing reliable analog circuits in an unreliable world. Proceedings of the IEEE 2012 Custom Integrated Circuits Conference, 2012, pp. 1–4. doi:10.1109/CICC.2012.6330571.
34. Puzzi, S.; Carpinteri, A. A double-multiplicative dynamic penalty approach for constrained evolutionary optimization. *Structural and Multidisciplinary Optimization* **2008**, *35*, 431–445.
35. Dehbashian, M.; Maymandi-Nejad, M. Co-AGSA: An efficient self-adaptive approach for constrained optimization of analog IC based on the shrinking circles technique. *Integration* **2017**, *59*, 218–232. doi:https://doi.org/10.1016/j.vlsi.2017.06.003.
36. Shi, Y.; Liu, H.; Gao, L.; Zhang, G. Cellular particle swarm optimization. *Information Sciences* **2011**, *181*, 4460–4493.
37. Lagos-Eulogio, P.; Seck-Tuoh-Mora, J.C.; Hernandez-Romero, N.; Medina-Marin, J. A new design method for adaptive IIR system identification using hybrid CPSO and DE. *Nonlinear Dynamics* **2017**, *88*, 2371–2389.
38. Seck-Tuoh-Mora, J.C.; Hernandez-Romero, N.; Lagos-Eulogio, P.; Medina-Marin, J.; niga Peña, N.S.Z. A continuous-state cellular automata algorithm for global optimization. *Expert Systems with Applications* **2021**, *177*, 114930. doi:https://doi.org/10.1016/j.eswa.2021.114930.
39. Seck-Tuoh-Mora, J.C.; Hernandez-Romero, N.; nos, F.S.B.; Volpi-Leon, V.; Medina-Marin, J.; Lagos-Eulogio, P. A majority-minority cellular automata algorithm for global optimization. *Expert Systems with Applications* **2022**, *203*, 117379. doi:https://doi.org/10.1016/j.eswa.2022.117379.

40. Suganthan, P.N.; Hansen, N.; Liang, J.J.; Deb, K.; Chen, Y.P.; Auger, A.; Tiwari, S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL report* **2005**, *2005005*, 2005.
41. Deb, K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* **2000**, *186*, 311–338.
42. Ma, H.; Simon, D. Blended biogeography-based optimization for constrained optimization. *Engineering Applications of Artificial Intelligence* **2011**, *24*, 517–525.
43. Kruiskamp, W.; Leenaerts, D. DARWIN: CMOS opamp Synthesis by means of a Genetic Algorithm. 32nd ACM/IEEE Design Automation Conference, 1995, pp. 433–438.
44. Vural, R.; Erkmen, B.; Bozkurt, U.; Yildirim, T. CMOS Differential amplifier area optimization with evolutionary algorithms. World Congress on Engineering and Computer Science, WCECS, 2013, Vol. 2, pp. 666–670. San Francisco, USA.
45. Kameyama, K. Particle Swarm Optimization - A Survey. *IEICE Transactions on Information and Systems* **2009**, *E92D*, 1354–1361.
46. Jordehi, A.R. Particle swarm optimisation for dynamic optimisation problems: a review. *Neural Computing & Applications* **2014**, *25*, 1507–1516.
47. Zhang, Y.; Wang, S.; Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Mathematical Problems in Engineering* **2015**, *2015*, 38 pages.
48. Wolfram, S. *A New Kind of Science*; Wolfram Media Inc, 2002.
49. McIntosh, H.V. *One Dimensional Cellular Automata*; Luniver Press, 2009.
50. Adamatzky, A.; Martínez, G.J. *Designing Beauty: The Art of Cellular Automata*; Vol. 20, Springer, 2016.
51. Gao, L.; Huang, J.; Li, X. An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process. *Applied Soft Computing* **2012**, *12*, 3490–3499.
52. Gholizadeh, S. Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. *Computers & Structures* **2013**, *125*, 86–99.
53. Li, X.; Gao, L.; Wen, X. Application of an efficient modified particle swarm optimization algorithm for process planning. *The International Journal of Advanced Manufacturing Technology* **2013**, *67*, 1355–1369.
54. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Applied Intelligence* **2021**, *51*, 1531–1551.
55. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future generation computer systems* **2019**, *97*, 849–872.
56. Baykasoğlu, A.; Akpinar, Ş. Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems–Part 1: Unconstrained optimization. *Applied Soft Computing* **2017**, *56*, 520–540.
57. Seck-Tuoh-Mora, J.C.; Lopez-Arias, O.; Hernandez-Romero, N.; Martínez, G.J.; Volpi-Leon, V. A New Algorithm Inspired on Reversible Elementary Cellular Automata for Global Optimization. *IEEE Access* **2022**, *10*, 112211–112229.
58. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-based systems* **2020**, *195*, 105709.
59. Valencia-Ponce, M.A.; Tlelo-Cuautle, E.; de la Fraga, L.G. On the Sizing of CMOS Operational Amplifiers by Applying Many-Objective Optimization Algorithms. *Electronics* **2021**, *10*.
60. Maji, K.B.; De, B.P.; Kar, R.; Mandal, D.; Ghoshal, S.P. CMOS Analog Amplifier Circuits Design Using Seeker Optimization Algorithm. *IETE Journal of Research* **2022**, *68*, 1376–1385.
61. Paul, A.; Ramírez-Angulo, J.; Sánchez, A.D.; López-Martín, A.J.; Carvajal, R.G.; Li, F.X. An Enhanced Gain-Bandwidth Class-AB Miller op-amp With 23,800 MHz·pF/mW FOM, 11-16 Current Efficiency and Wide Range of Resistive and Capacitive Loads Driving Capability. *IEEE Access* **2021**, *9*, 69783–69797. doi:10.1109/ACCESS.2021.3077532.