SRTTU

Journal of Electrical and Computer Engineering Innovations

(JECEI)

Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

Low-Area/Low-Power CMOS Op-Amps Design Based on Total Optimality Index Using Reinforcement Learning Approach

N. Sayyadi Shahraki, S. H. Zahiri *

Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran

Article Info

Article History:
Received 07 July 2017
Revised 10 January 2018
Accepted 16 May 2018

Keywords:
Low-Area and Low-Power
CMOS op-amp
Multi-objective optimization
Reinforcement learning
Total optimality index

*Corresponding Author's Email Address: hzahiri@birjand.ac.ir

Extended Abstract

Background and Objectives: Today, the use of methods derived from Reinforcement learning-based approaches, due to their powerful in learning and extracting optimal/desirable solutions to various problems, shows a significant wideness and success. This paper presents the application of reinforcement learning in automatic analog integrated circuit design.
Methods: In this work, the multi-objective approach by learning automata is evaluated for accommodating required functionalities and performance specifications considering optimal minimizing the MOSFETs area and power consumption for two famous CMOS op-amps.
Results: The performance of the circuits is evaluated through HSPICE and the approach is implemented in MATLAB, so a combination of MATLAB and HSPICE is performed. The two-stage and single-ended folded-cascode op-amps are designed in 0.25µm and 0.18µm CMOS technologies, respectively. According to the simulation results, a power of 560.42 μW and an area of 72.825 μm^2 are obtained for a two-stage CMOS op-amp, and also a power of 214.15 μW and an area of 13.76 μm^2 are obtained for a single-ended folded-cascode op-amp. In addition, in terms of total optimality index, MOLA for both cases has the best performance between the applied methods, and other research works with values of -25.683 and -34.162 dB, respectively.
Conclusion: The results shown the ability of the proposed method to optimize aforementioned objectives, compared with three multi-objective well-known algorithms.

Introduction

Analog circuits are essential role in the design of integrated circuits (ICs), so that their importance cannot be ignored. Due to the complexity of analog circuits, their desirable manual design is difficult to obtain at a reasonable time. The analog circuit design process mainly includes the selection of the best topology, circuit sizing, and layout synthesis [1]-[4]. Deterministic and meta-heuristic optimization techniques are used to circuit sizing. Deterministic optimization algorithms involve methods such as Newton and Levenberg-Marquardt that provide the problems of choosing a good starting point, being trapped in the local optimum, and dependency on the continuity and differentiability of the cost functions [5]. Meta-heuristic algorithms, unlike the deterministic optimization methods, have high performance and ability for solving optimization problems [6]-[9]. The purpose of meta-heuristic algorithms is to find proper values for the decision design parameters of an optimization problem to optimize one/multiple cost function [10]. Designing the size of analog circuits is an important issue that can be carried out with innovative algorithm methods. Intelligent methods are being developed to design the size of analog circuits. With the advancement of ICs manufacturing technology, it is important to design circuits with high accuracy and in the smallest size possible.

Heuristic-based approaches make circuit design in the form of the single-objective (SO) and multi-objective (MO) optimization. Because circuits have conflicting goals, it is best to use multi-objective optimization (MOO). Unlike SO optimization, these methods attempts to find non-dominated solutions to optimize circuit performance. Several studies have been carried out in the design and optimization of circuits, which have achieved favorable results by providing approaches based on circuit theory and intelligent optimization techniques [11]-[25].

In Table 1, some of the recent similar research works are listed. In these studies, there are some defects for example, considering the problem as a SO, ignoring the execution time of the algorithms, not reporting the results of design parameters, etc. In addition to these methods, reinforcement learning can be considered as one of the other techniques used to design CMOS operational amplifiers (op-amps).

Table 1 : A review of the latest similar research

Ref.	Year	Technology	Cost function	Method
Barros et al., [13]	2010	UMC 0.18µm	Area, Power	GENOM
Vural et al., [14]	2012	TSMC 0.35 µm	Area	PSO
Mallick et al., [16]	2017	TSMC 0.35µm	Area	GSA_PSO
Dehbashian et al., [17]	2017	0.18 & 0.25 µm	Area, Power	Advanced GSA_PSO
Dehbashian et al., [18]	2017	0.25 & 0.18 µm	Area, Power	Co-AGSA

Learning automata (LA) is based on reinforcement learning and is one of the main components in adaptive learning systems. It is an important research area of artificial intelligence (AI) and has a wide range of applications [26]-[32]. For the first time, Tsetlin created a new model of computer learning in 1961, now known as a LA [33]. The main goal was to determine the optimal action of a set of actions in such a way that it has the maximum probability of reward [34]. Various types of LA-based algorithms have been developed. In this work, we have used the MO version of learning automata (MOLA) method [35] for the automated design of CMOS op-amps (a two-stage CMOS op-amp and a single-ended folded-cascode op-amp).

Among the innovative aspects and contributions of the present work, we can mention: (1) The applicability of LA to optimize the CMOS Op-Amps; (2) The proper definition of the problem constraints and cost functions to achieve an desirable compromise between performance characteristics; (3) Providing a comprehensive criterion to evaluate proposed approach due to the simultaneous effect of objectives and design specifications on the optimization problem; (4) The performance numerical and statistical evaluations of the proposed approach by expressing visual and statistical indicators and employing powerful competitor algorithms.

The structure of the paper is as follows. Section 2 provides a brief review of MOO and other MOO algorithms. In Section 3, our proposed tool, case studies, cost functions, Pareto-front evaluation criteria, and the suggested index are introduced. Also, the simulation results of the proposed tool in the two study cases and their comparison with other algorithms are provided in Section 4. Finally, the conclusions are summarized in Section 5.

Multi-Objective Meta-Heuristic Optimization

In real applications, we usually deal with problems that under specific circumstances are faced with several cost functions simultaneously. These issues are in the field of MOO. In other words, the role of a MOO is to simultaneously optimize two or more cost functions. These objectives are usually in trade-off. In this method, unlike the SO method, which only receives an acceptable solution, there is a set of optimal solutions, known as Pareto-optimal solutions or Pareto-front. In such problems, a set of solutions, which complies each cost function with an acceptable level, is defined as optimal solutions. In this paper, for the first time, the MOLA method is used along with three rival MOO algorithms (called NSGA-II [36], MOPSO [37], and MOIPO [38]) and all of them are defined in the following subsections.

A. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

In NSGA-II, sorting and ranking all solutions are created by the main features (diversity, convergence, and robustness of solutions in the Pareto-front) in order to choose better solutions to create new offsprings. The NSGA-II is based on fast non-dominated sorting and crowding distance assignment methods. The procedure ensure elitism allows adding constraints to ensure that the solutions are feasible. The NSGA-II creates a population of individuals and then creates a non-domination level to rank and sort each individual. Then, it utilizes cross-over, mutation, and selection operators to produce new offspring. Subsequently, the parents and offspring are combined before partitioning the new combined pool into fronts [36]. The pseudo-code of the NSGA-II is shown by algorithm 1.

Algorithm 1: The pseudo-code of NSGA-II

I. Initialize the parameters of the algorithm.
II. Initialize the population *nPopi*, , $1 \leq i \leq N$.
III. Evaluate the cost functions.

IV. Assign rank (level) based on Pareto dominance sort.
V. Generate child population.
VI. Perform binary tournament selection, recombination and mutation.
VII. **for** all *nPopi*, **do**
VIII. **for** each parent and child in population, **do**
IX. Assign rank (level) based on Pareto sort.
X. Generate sets of non-dominated vectors along PF_{known} (known/current Pareto-front).
XI. Adding solutions to the next generation starting from the first front until *N* individuals are found determine crowding distance between points on each front.
XII. end for
XIII. Select points (elitist) on the lower front (with lower rank) and outside a crowding distance.
XIV. Create next generation.
XV. Perform binary tournament selection, recombination and mutation.
XVI. end for
XVII. **return** Pareto-front solutions

B. Multi-Objective Particle Swarm Optimization (MOPSO)

The PSO is one of the most important intelligent optimization algorithms. The main idea of this algorithm is taken from the social behavior of animals, such as birds and fish. Therefore, the search factor in this algorithm is birds and fish called particles. It is accompanied by a massive movement of birds and a shift in direction to their best position and their neighbors [39]. One of the most popular and effective proposals for MO versions of the PSO optimization algorithm is presented in [37]. The pseudo-code of the MOPSO is shown by algorithm 2.

Algorithm 2: The pseudo-code of MOPSO

I. Initialize the parameters of the algorithm, maximum iteration (*MaxIt*).
II. Initialize the population *nPopi*, $1 \le i \le N$.
III. Determine the initial speed (v_i=0).
IV. Evaluate the cost functions.
V. Store the positions of the particles that represent non-dominated vectors in the repository.
VI. Generate hyper cubes.
VII. **while** *MaxIt* has not reached, **do**
VIII. Calculate the speed of each particle [39].

$$\begin{aligned} v_i^d(t+1) &= \omega(t) \cdot v_i^d(t) \\ &+ C_1 \cdot rand_1(pbest_i^d(t) - x_i^d(t)) \\ &+ C_2 \cdot rand_2\,(gbest^d(t) - x_i^d(t)) \end{aligned} \quad (1)$$

where $\omega(t)$ is inertia weight, $pbest_i^d$ is personal best position particle *i* in iteration *t*, $gbest^d$ is the global best position *i* in the *d*-th dimension, C_1 and C_2 are cognitive and social parameters.
IX. Calculate the position and return the particles that are out of range of search to the search space, according to (2).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(\text{t+1}) \quad (2)$$

X. Evaluation of cost functions.
XI. Update repository (add non-dominant particles and eliminate dominated particles)
XII. Control the volume of the repository.
XIII. end while
XIV. **return** Pareto-front solutions

C. Multi-Objective Inclined Planes system Optimization (MOIPO)

The search factors in inclined planes system optimization (IPO) algorithm are the number of small balls that are located on a sloping surface without friction. Three attributes of position, height and angels in relation to other balls are considered for each ball. The main idea of this algorithm is to assign a height to each ball according to its objective function. Height values represent the potential energy of the balls, and the movement of the balls downwards converts potential energy to kinetic energy and causes acceleration. In fact, agents tend to tine their potential energy and to reach the minimum point (s). The position of each agent is a possible solution in the problem space [40]. The MO version of the algorithm has been created in [38]. Also, the algorithm 3 shows its pseudo-code.

Algorithm 3: The pseudo-code of MOIPO

I. Initialize the parameters of the algorithm, maximum iteration (*MaxIt*).
II. Initialize the population *nPopi*, $1 \le i \le N$.
III. Determine the initial speed and acceleration (v_i=0, a_i=0)
IV. Evaluate the cost functions.
V. Store the positions of the balls that represent non-dominated vectors in the repository.
VI. Generate hyper cubes.
VII. **while** *MaxIt* has not reached, **do**
VIII. Calculate the acceleration and speed of each ball [38].

$$a_i^d(t) = \sum_{j=1}^{N} U(f_j(t) - f_i(t)) \cdot \sin(\phi_{ij}^d(t)) \quad (3)$$

$$v_i^d(t) = \frac{x_{best}^d(t) - x_i^d(t)}{\Delta t} \quad (4)$$

where $U(.)$ is the unit step function, $\phi_{i,j}^d$ is angle between the *i*-th ball and *j*-th ball in *d*-th dimension.
IX. Calculate the position and return the balls that are out of range of search to the search space [38].

$$\begin{aligned} x_i(t+1) &= k_1 \cdot rand_1 \cdot a_i^d(t) \cdot \Delta t^2 \\ &+ k_2 \cdot rand_2 \cdot v_i^d(t) \cdot \Delta t + x_i^d(t) \end{aligned} \quad (5)$$

$$k_1(t) = \frac{c_1}{1 + \exp((t - shift_1) \cdot scale_1)} \quad (6)$$

$$k_2(t) = \frac{c_2}{1 + \exp(-(t - shift_2) \cdot scale_2)} \quad (7)$$

where c_1, c_2, $shift_1$, $shift_2$, $scale_1$, and $scale_2$ are constants determined for each function, experimentally.
X. Evaluate the cost functions.
XI. Update repository (add non-dominant balls and eliminate dominated balls).
XII. Control the volume of the repository.
XIII. end while
XIV. **return** Pareto-front solutions

D. Multi-Objective by Learning Automata (MOLA)

The MOLA is found more practicable and efficient in finding accurate solutions for complex optimization problems. The number of automata used in the MOLA algorithm is equal to the dimension of the problem. For *N* dimensional problem, the MOLA includes *N* automata [35]. Each automaton is responsible for searching one dimension and acts independently in the environment. The pseudo-code of the MOLA is listed in algorithm 4.

Algorithm 4: The pseudo-code of MOLA

XV. Initialize the parameters of the MOLA, maximum iteration (N_{femax}), set N_{fe}=0.
XVI. Initialize a state $X=[x_1,\ldots,x_i,\ldots,x_N]$, randomly.
XVII. Determine the initial cell value randomly at the rang of [0,1].
XVIII. Evaluate the cost functions.
XIX. Store the positions of the actions that represent non-dominated vectors in the repository.
XX. Generate hyper cubes.
I. **while** ($N_{fe} \leq N_{femax}$), **do**
II. **for** i=1 to N, **do**
III. Select an unselected dimensional state x_i randomly.
IV. Estimate path values $L_l(x_i)$, according to (8) [35].

$$L_l(x_i) = (1-\lambda_1)\sum_{m-1}^{k-1} \lambda_1^{m-1}\cdot v^*_{l,m} + \lambda_1^{k-1}\cdot v^*_{l,k} \tag{8}$$

V. Calculate p_1 (probability of selection the left path or the right path) and p_2 (probability of choosing a cell between the k cells located on the path determines) [35].

$$p_1(L_l(x_i)) = \frac{exp(\frac{L_l(x_i)}{\tau})}{\sum_{m=\tau}^{2} exp(\frac{L_m(x_i)}{\tau})} \tag{9}$$
$$l = 1,2$$

$$p_2(c_{i,j+s}) = \frac{exp(\frac{V(x_i)|_{x_i\in c_{i,j+s}}}{2\tau})}{\sum_{m=1}^{k} exp(\frac{V(x_i)|_{x_i\in c_{i,j+m}}}{2\tau})} \tag{10}$$
$$l = 1,2$$
$$s = 1,2,\ldots,k$$

where $V(x_i)$ is cell value. Temperature τ creates trade-off between the exploration and exploitation.

VI. Calculate reinforcement signal according to (11), set N_{fe}=N_{fe}+1, and update X_{best} (12) [35]

$$r(X(x_i')) = \begin{cases} 1 & if\ X(x_i')\ is\ a\ nondominated\ solution \\ 0 & otherwise \end{cases} \tag{11}$$

$$X_{best} \leftarrow \begin{cases} X(x_i') & if X(x_i')\ is\ a\ nondominated\ solution \\ X_{best} & otherwise \end{cases} \tag{12}$$

where, $X(x_i') = [x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots,x_N]$

VII. Update cell value $V(x_i)|_{x_i\in c_{i,j}}$ according to (13), where $j = floor(\frac{x_i - x_{min,i}}{\omega_{c,i}})$

$$V(x_i)|_{x_i\in c_{i,j}} \leftarrow r(X(x_i)) + \alpha_1.V(x_i)|_{x_i\in c_{i,j}} + (1-\alpha_1)((1-\lambda_2).L_{max}(x_i) + \lambda_2.L_{min}(x_i)) \tag{13}$$

VIII. Move action to the new cell with a step length η, according to (14). When the L_1 is selected, current dimensional state $x_i$moves to $x_i = x_i - \eta$ and with the choice of L_2, x_i moves to $x_i = x_i + \eta$.

$$\eta = (\xi + rand).\omega_{c,i} \tag{14}$$

where ξ is distance (in the form of number of cells) between the current cell and the selected cell.

IX. Evaluate the cost functions.
X. Store the positions of the actions that represent non-dominated vectors in the repository.
XI. Generate hyper cubes.
XII. **while** r(X)==1, **do**
XIII. Go to step (X)
XIV. **end while**
XV. **end for**
XVI. Use (15) to add perturbations to the dimensional states of X_{best}, use (12) to update X_{best}, set N_{fe}=N_{fe}+1.

$$X' \leftarrow X + \Delta + rand.(X_{best} - X) \tag{15}$$

where Δ_i is $\Delta_i = sign(\kappa).\zeta.(x_{max,i} - x_{min,i})$ and κ is $\kappa = V(y_i)\,|_{y_i\in c_{i,j+1}} - V(z_i)\,|_{z_i\in c_{i,j-1}}$.The ζ is random variable ($\zeta \in [0,\frac{k}{D}]$). Also, $sign(\kappa)$ is a sign function.

XVII. **end while**
XVIII. **return** Pareto-front solutions

An Automated Design Simulation Tool of CMOS Analog ICs Based on Circuit Intelligent Sizing for Low-Power/Low-Area Using the Proposed MOLA Method

In this Section, an optimization tool is proposed for automated design of analog IC. It should be noted that analog circuits are simulated by HSPICE simulator. Also, MOLA and MO algorithms are performed in MATLAB. By connecting MATLAB and HSPICE software, the optimization process is done (Fig. 1). In the beginning, design parameters and design specifications are determined by the designer, while a reasonable predefined range is also taken into account for each design parameter. Note that design parameters consist of the length and width of the CMOS transistors, capacitor values, and biasing currents.

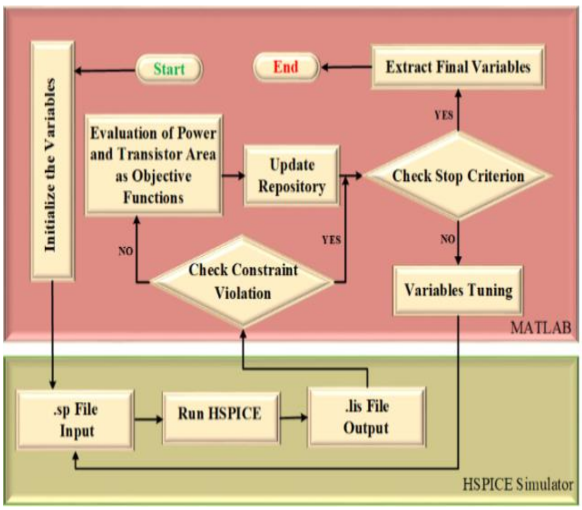


Fig. 1: The general structure of the automated circuit design simulation tool based on the intelligent MOO.

Continue on this section, the desired amplifier circuits, cost functions, the proposed index, and Pareto-front evaluation criteria are explained.

A. Case Studies

In this paper, in order to show the performance of MOLA method in the design of analog circuits, two important amplifiers are used. A two-stage CMOS op-amp was designed in 0.25μm technology and a single-ended folded-cascode op-amp with 0.18μm technology. There are 13 design parameters in each circuit.

In Fig. 2, a two-stage CMOS op-amp is shown with miller. Design parameters in this circuit include transistor widths and lengths, biasing current (I_{bias}), compensation capacitance (C_c), and load capacitance (C_L). Here, the appropriate matching relations are also imposed as M_1=M_2, M_3=M_4, and M_5=M_8. Furthermore, the positive power supply (V_{DD}) and the negative power supply (V_{SS}) are equal to 2.5V and -2.5V, respectively [18]. This circuit set values for the C_C and C_L that provide C_C>0.22C_L [16].

Fig. 3 shows a single-ended folded-cascode amplifier. In this case, the biasing current, the total values of transistor widths and lengths are considered as design parameters. Here, the appropriate matching relations are also imposed as M_1=M_2, M_3=M_4=M_{bp}, M_5=M_{bn}, M_6=M_7, M_8=M_9, and M_{10}=M_{11}. Furthermore, the positive power supply (V_{DD}), the negative power supply (V_{SS}), and C_L are equal to 0.9V, -0.9V and 5pF, respectively [18].

Desired specifications (small-signal differential voltage gain (DC gain), gain-bandwidth product (GBW), phase margin, output swing, common mode rejection ratio (CMRR), and power supply rejection ratio (PSRR), etc.) are in accordance with Table 2 for both cases. Also, some equations are consistent with the circuit according to the following equations [18].

$$dm_i = \frac{V_{DS}(M_i)}{V_{DS_{sat}}(M_i)} \tag{16}$$

$$OSP = V_{DD} - abs\,(V_{DS_{sat}}(M_{10})) - abs\,(V_{DS_{sat}}(M_4)) \tag{17}$$

$$OSN = V_{SS} - abs\,(V_{DS_{sat}}(M_8)) - abs\,(V_{DS_{sat}}(M_6)) \tag{18}$$

where, dm_i is the ratio of the drain-source voltage (V_{DS}) to the saturated drain-source voltage ($V_{DS_{sat}}$) of i-th transistor (M_i), and *OSP* and *OSN* are the output swing positive and output swing negative, respectively.

B. Cost Functions

Herein, intelligent methods are used to optimize the two important and essential indicators of amplifier circuits, namely power consumption and total area MOSFETs. This cost functions are considered as follows.

- Minimizing power consumption
- Minimizing the total area MOSFETs

C. Total Optimality Index (TOI)

In this paper, due to the diversity and multiplicity of qualitative indicators in the design problem, a total index is presented that illustrates the success of the optimization method. This indicator can be used to investigate the performance of the proposed optimization algorithm in the design problem. Therefore, a criterion called total optimality index (TOI) is proposed.

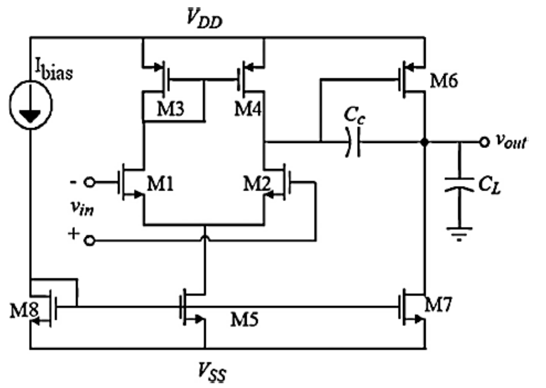


Fig. 2: The two-stage CMOS op-amp circuit (Case 1) [18].

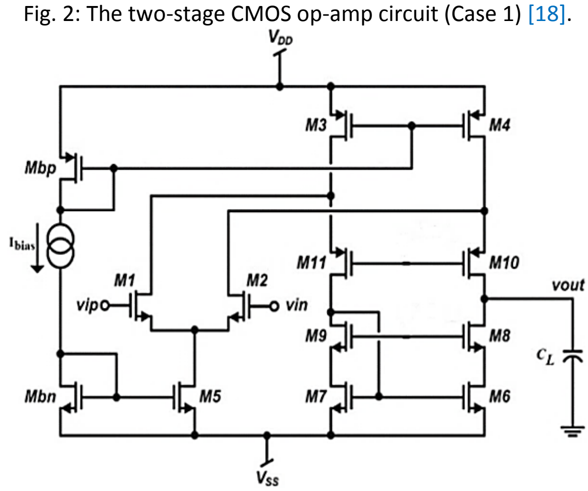


Fig. 3: The single-ended folded-cascode op-amp circuit (Case 2) [18].

Table 2 : Desired characteristics of Case 1 and Case 2

Design specification	Constraint	
	Case 1	Case 2
DC gain (dB)	≥70	≥70
GBW (MHz)	≥2	≥12
Phase Margin (deg)	≥50	≥55
Slew Rate (V/μs)	≥1.5	≥10
Output Swing (V)	≥2	—
CMRR (dB)	≥70	—
$PSRR^+$ (dB)	≥70	—
$PSRR^-$ (dB)	≥70	—
dm_1, dm_3, dm_5, dm_6, dm_8, dm_{10}	—	≥1.2
OSP (V)	—	≥0.5
OSN (V)	—	≤-0.5
MOSFETs	Saturation	Saturation

The TOI has been introduced to express the impact of the design specifications and the cost functions of the problem. The lower value of TOI represents the more favorable response. The index is defined as follows:

$$TOI = 20\ log_{10}(norm(A(\mu m^2)).P(mW).\frac{\sum abs(C_B)}{\sum abs(C)}) \quad (19)$$

where, A is total area of the MOSFET in µm^2 and P is the power consumption in mW (as the cost functions), C is design specifications (constraints) and C_B is specifications boundary value in the problem of designing an amplifier circuit. In (19), to balance the values of power and area, the amount of area is normalized between zero and one, and due to the negativity of some of the design specifications; the absolute values of C and C_B are used. Also, with a mean less than a fractional value based on this theory and dominant experimental results, the final TOI was logarithmically considered. The most desirable TOI (minimum) is created by minimizing cost functions and maximizing the design specifications. The design of the TOI is such that the main focus is on the cost functions of the problem and a minor improvement in one of them will minimize the TOI. The logarithm coefficient of 20 is to increase the TOI sensitivity to a partial improvement of the fitness values of the cost functions.

D. Pareto-front evaluation criteria

In order to evaluate Pareto-front, two criteria of the overall non-dominated vector generation (ONVG) and spacing (SP) are used. Despite the existence of other criteria for studying the quality of the Pareto-front, the reason for choosing these two criteria is that there is no need to know the real Pareto-front and they are produced in accordance with the received Pareto-front.

- ONVG: The ONVG represents the number of optimally non-dominated responses (based on Pareto-front) in a MO problem [38].

$$ONVG \equiv |PF_{known}| \quad (20)$$

where, $|PF_{known}|$ is the number of vectors in PF_{known}.

- SP: The SP numerically represents the spread of the vectors in the PF_{known} and measures the distance variance of neighboring vectors in it (as (21)) [38].

$$SP = \sqrt{\frac{\sum_{i=1}^{n}(\bar{d} - d_i)^2}{n-1}} \quad (21)$$

where, $d_i = min_j(|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$, $i,j = 1,2,\dots,n$. The $\bar{d}$ is the mean of all d_i, and n is the number of vectors in PFknown ($|PF_{known}|$). So that, $SP = 0$, means that all members are spaced evenly apart in [38].

Simulation and Analysis Results

This section is divided into two subsections. In the first subsection, the results and analysis are presented in the optimization of two circuits. All the results are reported in the form of the values of cost functions, design parameters, design specifications, and TOI. The best, worst, mean, and variance of the values of the cost functions and the TOI are presented for proposed method in the best run. The integration index, Pareto-front, and bode curve figures are also presented. The second subsection is organized by the qualitative and quantitative analysis of the proposed MOLA approach. In addition, the Pareto indexes and the runtime of MOLA performance are analyzed in comparison with other proposed algorithms in the best run. All implementations are performed in MATLAB 2016a MathWorks and HSPICE A-2008.3 under a computer system with Intel® Core™ i5-4460U CPU @ 3.20GHz, 4GB RAM, and Windows Enterprise 10. The vectors of design parameters that should be determined by the proposed tools are as follows:

$$X_{i\_Case1} = [W_1, W_3, W_5, W_6, W_7, L_1, L_3, L_5, L_6, L_7, C_L, C_C, I_{bias}] \quad (22)$$

$$X_{i\_Case2} = [W_1, W_3, W_5, W_6, W_8, W_{10}, L_1, L_3, L_5, L_6, L_8, L_{10}, I_{bias}] \quad (23)$$

The details on design parameters for both cases are listed in Table 3. Also, all control parameters of the proposed methods are presented in Table 4.

Table 3: The range of design parameters

Design parameter	Case 1		Case 2	
	Lower bound	Upper bound	Lower bound	Upper bound
W (µm)	5	40	0.24	200
L (µm)	0.25	2	0.18	5
I_{bias} (µA)	20	40	30	400
C_C (pF)	2	20	—	—
C_L (pF)	7	15	—	—

A. Results and Comparisons for Case Studies

In order to demonstrate the ability of the reinforcement learning method to solve the problem of circuit optimization, the results are compared with several intelligent methods and previous studies.

Table 5 and Table 6 show the best solution in the best run of the algorithms for the case 1 and case 2 that is generated by the best TOI (minimum). The bolded responses show the best values in terms of design specifications, cost functions, and TOI. In Case 1, the MOLA method is able to produce the low-area and low-power with the values 72.825 µm^2 and 560.42 µW, respectively, and these values are created with the minimum values for design parameters such as biasing current, compensation capacitance, and load capacitance. The algorithms intelligently set values for the C_C and C_L that provide C_C>0.22C_L. Additionally, the ability of the MOLA is more specific than other algorithms in the TOI. According to the table, this superiority, relative to the best Pareto-optimal solution of algorithms, is achieved with 56.14%, 12.5%, 8.33%, and 25% by MOLA, MOPSO, MOIPO, and NSGA-II, respectively. Table 6 shows that from the point of view of area assessment, the MOLA method for Case 2 is superior to MOIPO, MOPSO and NSGA-II with the values of 67.45%, 65.25%, and 32.85%, respectively. The results show that the MOLA method with the least power and area can produce the best TOI with a value of -34.1627. Also, Table 7 shows the statistical comparisons proposed method.

The evaluation of the proposed strategy for improving the integrated circuit criterion and the power consumption in the best run (minimum TOI) for case studies is shown in Fig. 4 and Fig. 5.

Fig. 6 shows the Pareto-fronts in the best run (in terms of TOI criterion) for the proposed algorithms. The variety and the number of presented Pareto-front solutions provide a wide range of selection for the circuit designer. MOLA responses have dominated Pareto-front solutions of other algorithms. The HSPICE simulation results obtained from the optimally designed two op-amps are shown in Fig. 7 and Fig. 8 . Also, they show the values of Gain DC, GBW, Phase Margin, and Gain Margin for the MOLA in a solution which has the best TOI.

A comprehensive comparison between the results of the proposed MOLA algorithm and those of other rival methods is presented in Tables 8 and 9 along with other studies.

According to these tables, the circuit optimization by MOLA, compared to recent studies, has produced favorable values of power with the smallest MOSFET size.

It can be said that an effective trade-off between optimized low-power and low-area in the results of the proposed automata algorithm is more desirable.

Table 4: Control settings

NSGA-II	MOPSO	MOIPO	MOLA
Total Run=20	*Total Run*=20	*Total Run*=20	*Total Run*=20
$MaxIt_{Case1}$=100	$MaxIt_{Case1}$=100	$MaxIt_{Case1}$=100	$N_{femax\ Case1}$=2000
$MaxIt_{Case2}$=150	$MaxIt_{Case2}$=150	$MaxIt_{Case2}$=150	$N_{femax\ Case2}$=3500
nPop=20	*nPop*=20	*nPop*=20	*nRep*=20
nRep=20	*nRep*=20	*nRep*=20	*D*=500
P_c=0.9	c_1=1.4962	c_1=0.1	*k*=50
P_m=0.1	c_2=1.4962	c_2=3.05	$\alpha_1 \in [0,1]$
$\eta_c = 2$	*W*=1	$shift_1$=100	$\lambda_1 = 0.5$
$\eta_m = 18$	W_{damp}=0.73	$shift_2$=300	$\lambda_2 \in [0,1]$
—	$nGrid_{Case1}$=4	$scale_1$=0.03	$\tau \in [0,0.5]$
—	$nGrid_{Case2}$=10	$scale_2$=0.03	$nGrid_{Case1}$=4
—	$\alpha = 0.1$	$nGrid_{Case1}$=4	$nGrid_{Case2}$=10
—	$\beta = 4$	$nGrid_{Case2}$=10	$\alpha = 0.1$
—	$\gamma = 2$	$\alpha = 0.1$	$\beta = 4$
—	—	$\beta = 4$	$\gamma = 2$
—	—	$\gamma = 2$	—

B. Performance Analysis

Tables 10-14 report statistical analysis of the performance of the MOLA method in relation to the control parameters (*k*, *D*, and N_{femax}) on the cost functions and TOI in the best run (based on TOI value). In Table 13, the effect of the MOLA parameters on the implementation time and Pareto-front criterion is reported.

Finally, for the performance analysis of Pareto indexes and runtime of MOLA with other assumed algorithms in the best run, Table 14 is provided.

Table 5: Optimal results of parameters based on the best solution for all methods (Case 1)

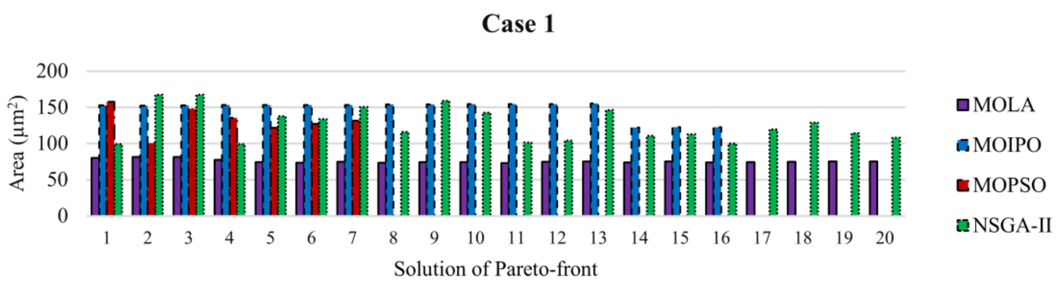
Parameter		The best Pareto-solution/method			
		20/NSGA-II	16/MOIPO	4/MOPSO	1/MOLA
Design parameter	W_1/L_1=W_2/L_2 (μm/μm)	7.184/1.336	9.999/**1.306**	6.484/1.416	**5.104**/1.654
	W_3/L_3=W_4/L_4 (μm/μm)	19.765/**0.524**	25.108/0.731	19.366/1.272	**6.813**/0.537
	W_5/L_5=W_8/L_8 (μm/μm)	12.408/**1.055**	15.825/1.408	13.649/1.26	**9.221**/1.479
	W_6/L_6 (μm/μm)	**24.584**/0.377	30.548/**0.341**	33.489/0.548	25.988/0.735
	W_7/L_7 (μm/μm)	15.694/0.872	23.706/1.58	16.599/0.919	**12.071/0.797**
	C_C (pF)	9.748	12.635	10.845	**7.831**
	C_L (pF)	9.094	14.138	9.181	**8.868**
	I_{bias} (μA)	21.77966	27.981	23.889	**20.037**
Design specification	DC gain (dB)	70.301	73.637	**78.197**	72.751
	GBW (MHz)	**2.525**	2.328	2.091	2.110
	Phase margin (deg)	**55.102**	50.164	54.470	50.596
	Slew rate (V/μs)	2.309	2.661	2.436	**3.020**
	Output Swing (V)	**2.412**	2.327	2.355	2.306
	CMRR (dB)	80.101	87.416	99.358	**104.156**
	$PSRR^+$(dB)	76.147	75.570	**85.245**	82.222
	$PSRR^-$ (dB)	88.012	103.460	**115.7**	82.892
Objective	Area (μm^2)	119.513	155.260	135.632	**80.199**
	Power consumption(μW)	678.47	748.39	641.09	**562.41**
TOI		-19.7231	-16.7304	-20.2187	**-25.5832**

Table 6: Optimal results of parameters based on the best solution for all methods (Case 2)

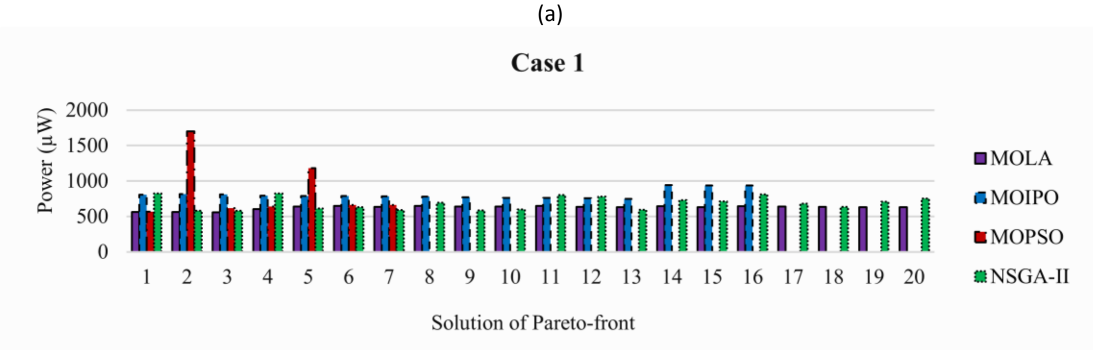
Parameter		The best Pareto-solution/algorithm			
		1/NSGA-II	14/MOIPO	7/MOPSO	1/MOLA
Design parameter	W_1/L_1 (μm/μm)	22.46/0.858	80.009/3.440	55.645/1.608	**13.27/0.515**
	W_3/L_3 (μm/μm)	76.544/0.504	**49.713**/0.351	116.105/0.396	84.054/**0.322**
	W_5/L_5 (μm/μm)	67.766/1.134	120.010/2.110	104.32/2.863	**19.103/0.57**
	W_6/L_6 (μm/μm)	5.21/1.691	6.254/2.449	10.793/2.859	**0.619/0.467**
	W_8/L_8 (μm/μm)	19.846/0.842	42.714/1.222	88.313/2.606	**18.057/0.732**
	W_{10}/L_{10}(μm/μm)	28.56/**1.068**	79.343/2.021	59.278/2.962	**16.956**/1.347
	I_{bias} (μA)	**61.519**	57.283	54.997	50.491
Design specification	DC gain (dB)	75.881	80.588	**82.502**	70.782
	GBW (MHz)	**15.507**	13.777	13.215	12.538
	Phase margin (deg)	82.067	**70.049**	60.925	83.077
	Slew rate (V/μs)	12.209	11.423	**10.973**	10.098
	dm_1	11.906	10.200	12.479	**14.480**
	dm_3	2.499	3.077	**3.316**	2.739
	dm_5	4.106	**4.286**	3.807	4.0319
	dm_6	1.644	**1.679**	2.138	1.514
	dm_8	**2.895**	2.943	2.084	**4.564**
	dm_{10}	5.032	**5.721**	4.952	3.664
	OSP (V)	0.620	**0.654**	0.635	0.576
	OSN (V)	-0.567	-0.563	**-0.607**	-0.555
Objective	Area (μm^2)	20.494	39.600	42.277	**13.760**
	Power consumption (μW)	277.36	256.54	244.290	**221.07**
TOI		-29.1398	-23.535	-23.0556	**-34.1627**

Table 7: Total statistical comparison of objectives and TOI for MOLA (Case 1 & Case 2)

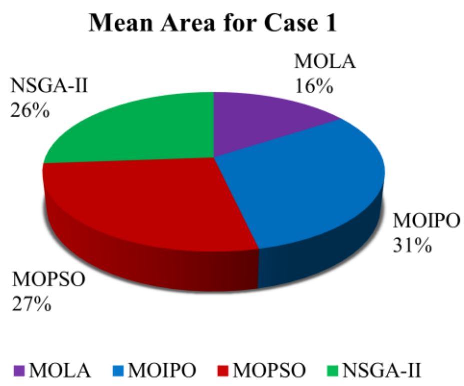
MOLA		Case 1	Case 2
	Area (μm^2)	72.825	13.759
Best	Power consumption (μW)	560.420	214.150
	TOI	-25.583	-34.163
	Area (μm^2)	81.563	14.924
Worst	Power consumption (μW)	650.350	221.070
	TOI	-24.0644	-33.848
	Area (μm^2)	75.499	14.458
Mean	Power consumption (μW)	625.590	215.064
	TOI	-25.0644	-33.848
	Area (μm^2)	6.663	0.084
Variance	Power consumption (μW)	857.920	2.641
	TOI	0.0432	0.026



(a)



(b)



(c)

Fig. 4: Evaluated area and power consumption in case1, (a): Area, (b): Power consumption, (c): Integration percentage.

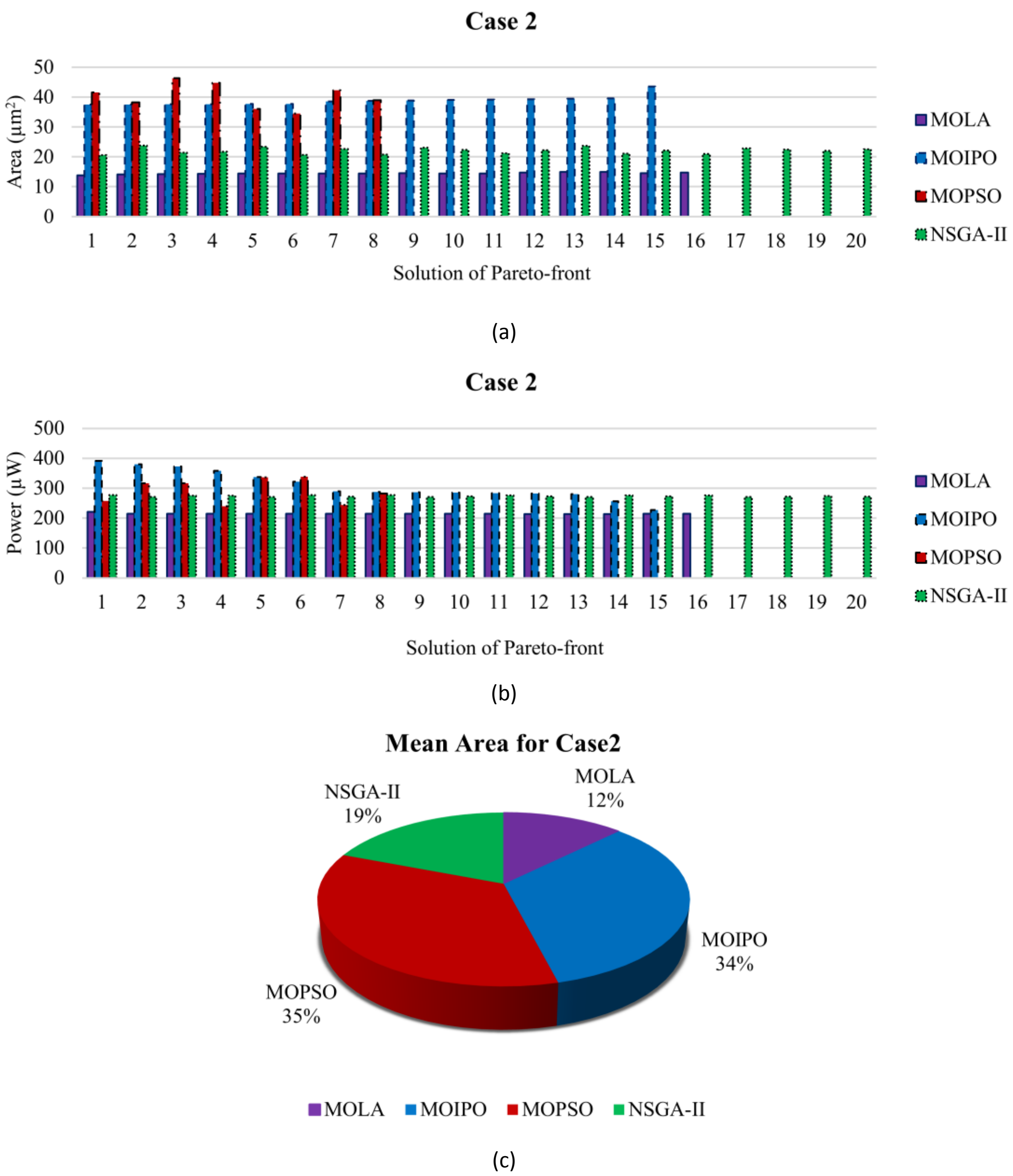
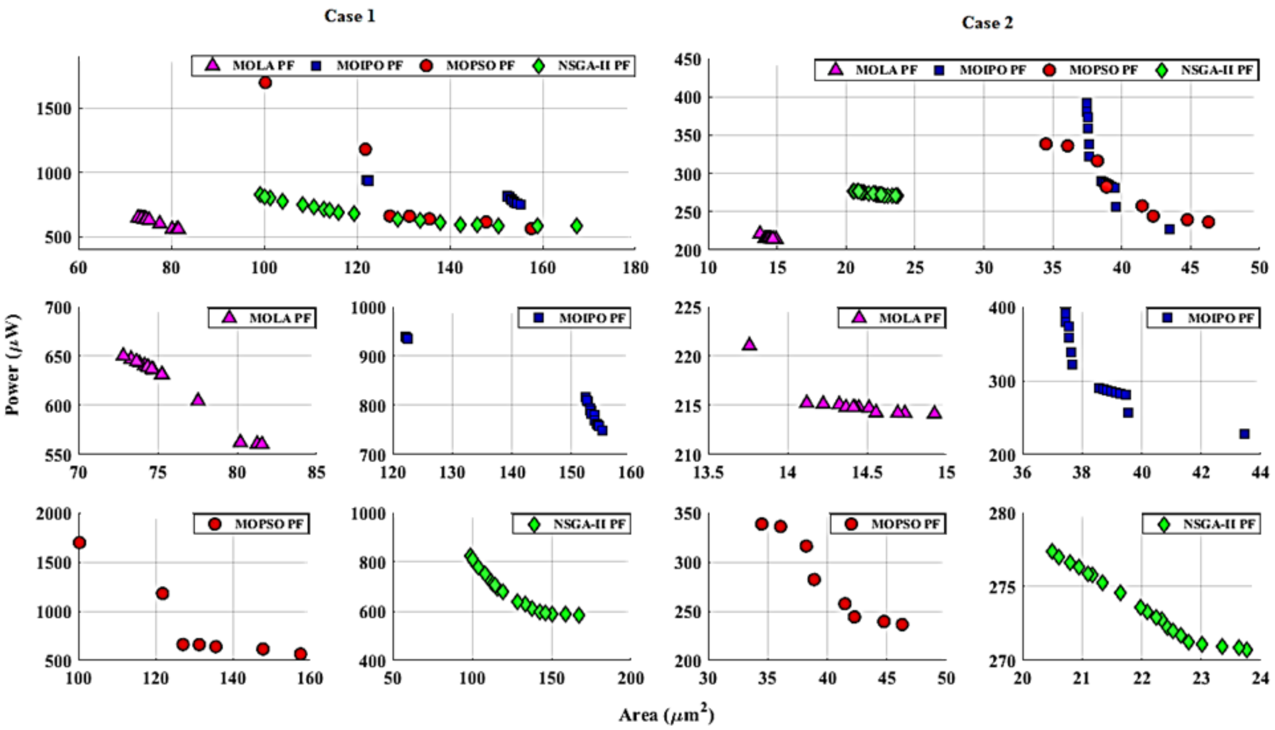


Fig. 5: Evaluated area and power consumption in case 2, (a): Area, (b): Power consumption, (c): Integration percentage.

Table 8 : Comparison with other methods (Case 1)

	PSO [14]	GSA-PSO [16]	AGSA_PSO+PF [17]	CO-GSA [18]	NSGA-II	MOPSO	MOIPO	MOLA
Technology (μm)	0.35	0.35	0.25	0.25	0.25	0.25	0.25	0.25
DC gain (dB)	42	75.43	70.441	74.785	71.024	**79.296**	75.63	73.808
GBW (MHz)	—	**5.776**	2.017	2.644	2.651	4.049	2.340	2.127
Phase margin (deg)	**83.8**	66.2	50.181	78.448	58.471	59.140	56.21	60.616
Slew rate (V/μs)	**22.4**	10.88	2.231	10.897	2.326	4.567	2.668	3.023
Output swing (V)	—	—	**2.415**	2.232	2.413	2.364	2.371	2.343
CMRR (dB)	84.2	87	88.187	78.040	**108.906**	108.677	87.416	104.156
$PSRR^+$ (dB)	40.1	83.2	72.675	**87.190**	76.403	86.172	83.413	82.530
$PSRR^-$ (dB)	68	110.4	**131.910**	86.650	88.421	121.630	103.460	95.485
Area (μm^2)	296	109.6	210.003	129.845	99.042	100.285	122.13	**72.825**
Power consumption (μW)	1260	712.8	700.710	**349.420**	582.670	565.930	748.390	560.420
TOI	-3.0968	-17.52	-8.1609	-12.3958	-19.7231	-20.2187	-16.7667	**-25.684**



(a) (b)

Fig. 6: Pareto-front of the MOIPO, MOPSO, NSGA-II, and MOLA, (a): Case 1, (b): Case 2.

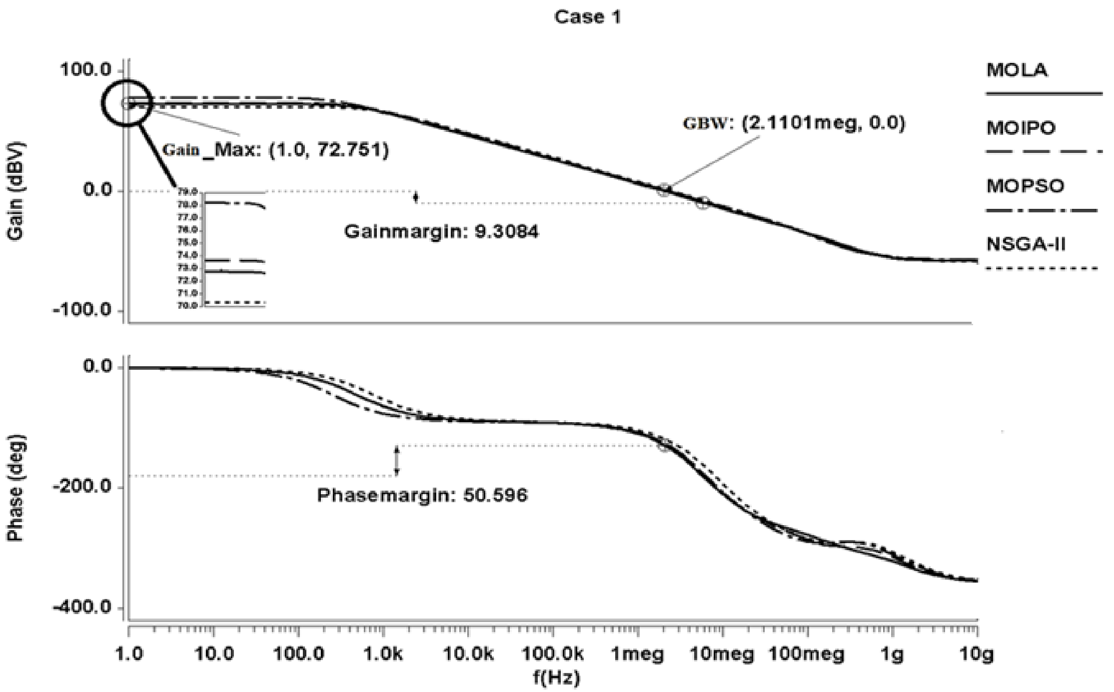


Fig. 7: Bode diagram plotted by the proposed methods in case 1.

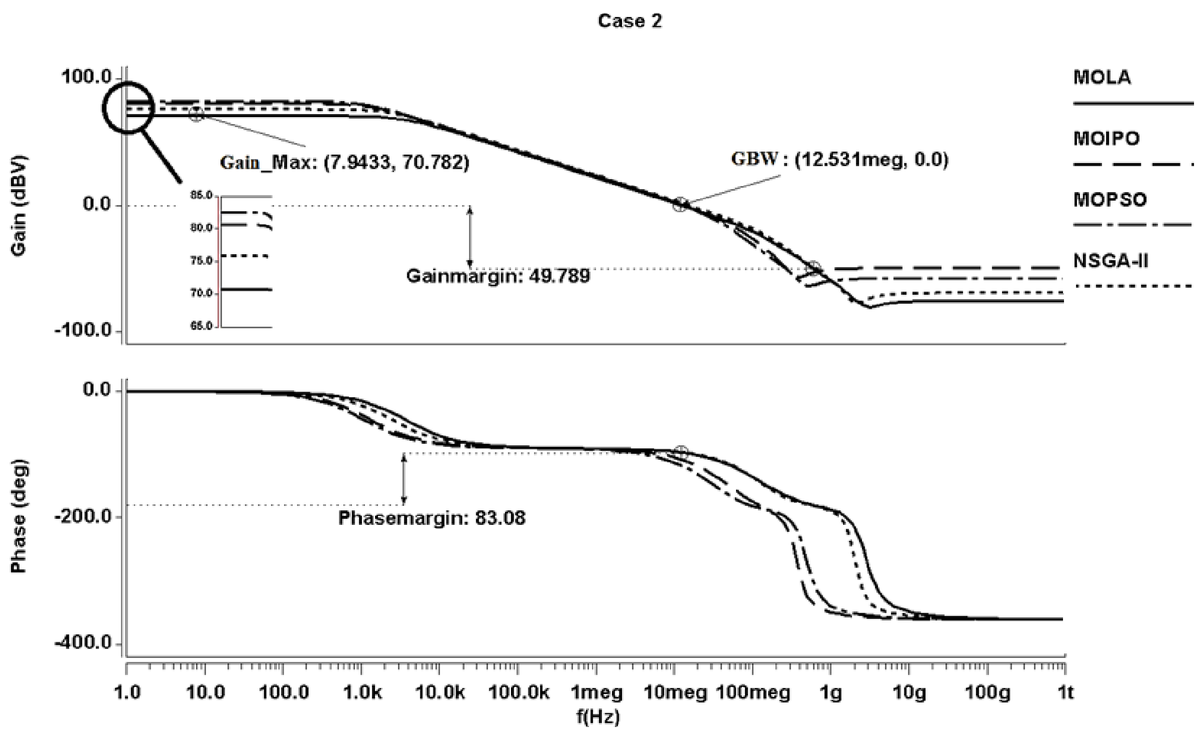


Fig. 8: Bode diagram plotted by the proposed methods in case 2.

Table 9: Comparison with other methods (Case 2)

	GENOM [13]	FRIDGE [18]	AGSA_PSO+PF [17]	NSGA-II	MOPSO	MOIPO	MOLA
Technology (µm)	0.18	0.18	0.18	0.18	0.18	0.18	0.18
DC gain (dB)	70.610	70.00	70.427	76.637	**85.349**	81.233	70.782
GBW (MHz)	15.350	16.00	15.505	15.507	16.359	**18.176**	12.538
Phase margin (deg)	79.60	80.60	**83.574**	82.067	68.032	70.049	83.077
Slew rate (V/µs)	15.360	15.300	10.001	12.212	15.400	**16.862**	10.098
dm_1	9.245	9.780	15.067	13.310	13.213	12.613	**16.571**
dm_3	1.568	**5.200**	2.419	2.499	3.733	3.257	2.739
dm_5	1.836	2.210	**5.540**	4.138	3.970	4.791	5.638
dm_6	8.171	10.500	2.217	1.971	3.216	**3.584**	1.513
dm_8	2.807	3.050	2.220	2.895	2.378	2.943	**4.564**
dm_{10}	1.653	1.950	**11.709**	5.032	5.818	5.721	3.664
OSP (V)	0.566	0.625	**0.695**	0.620	0.647	0.654	0.577
OSN (V)	-0.505	-0.502	-0.641	-0.606	-0.676	**-0.697**	-0.556
Area (μm^2)	16.870	23.710	16.961	20.494	34.500	37.439	**13.760**
Power consumption (µW)	244.60	233.30	222.10	270.73	236.61	227.47	**214.15**
TOI	-31.7005	-29.3704	-32.9950	-29.1398	-23.0556	-23.535	**-34.1627**

Table 10: Comparison of the effect of *k* on objectives and TOI (Case 2)

MOLA		(D=500, N_{femax}=3500)			
		k=4	k=10	k=50	k=100
Area (µm^2)	Best	28.817	21.4678	**13.760**	14.6994
	Worst	28.949	21.5047	**14.925**	16.8876
	Med	28.852	21.4833	**14.449**	16.5030
	Mean	28.846	21.4827	**14.458**	16.2131
	Variance	0.0008	**0.0001**	0.0840	0.5101
Power consumption (µW)	Best	427.040	307.180	**214.150**	247.870
	Worst	427.110	307.310	**221.070**	268.920
	Med	427.090	307.230	**214.770**	253.060
	Mean	427.094	307.237	**215.064**	257.799
	Variance	**0.0002**	0.0016	2.963**0**	49.0146
TOI	Best	-22.131	-27.252	**-34.424**	-32.044
	Worst	-22.091	-27.241	**-33.835**	-30.944
	Med	-22.120	-27.249	**-34.137**	-31.089
	Mean	-22.122	-27.248	**-34.119**	-31.220
	Variance	7.8698E-05	**8.9966E-06**	0.0249	0.1314

Table 11: Comparison of the effect of *D* on objectives and TOI (Case 2)

MOLA		(k=50, N_{femax}=3500)			
		D=100	D=500	D=1000	D=2000
Area (µm^2)	Best	14.894	**13.760**	19.503	26.802
	Worst	15.755	**14.925**	21.246	27.233
	Med	15.438	**14.449**	20.454	26.990
	Mean	15.426	**14.458**	20.4190	27.004
	Variance	0.057	0.0845	0.2522	**0.0152**
Power consumption (µW)	Best	228.420	**214.150**	221.9700	483.570
	Worst	230.380	**221.070**	223.1100	485.290
	Med	228.620	**214.770**	222.18	484.850
	Mean	228.847	**215.064**	222.3471	484.619
	Variance	0.3689	2.963	**0.1490**	0.2940
TOI	Best	-33.182	**-34.424**	-31.151	-20.951
	Worst	-32.595	**-33.835**	-30.381	-20.584
	Med	-32.836	**-34.138**	-30.705	-20.716
	Mean	-32.834	**-34.119**	-30.739	-20.732
	Variance	0.0247	0.0249	0.0506	**0.0186**

Table 12: Comparison of the effect of N_{femax} on objectives and TOI (Case 2)

MOLA		(k=50, D=500)			
		N_{femax}=1000	N_{femax}=3500	N_{femax}=5000	N_{femax}=10000
Area (µm^2)	Best	**12.333**	13.760	14.274	12.972
	Worst	15.106	14.925	15.463	**13.429**
	Med	**13.106**	14.449	14.734	13.388
	Mean	**13.171**	14.458	14.872	13.348
	Variance	0.5778	0.0845	0.1486	**0.0138**
Power consumption (µW)	Best	281.180	**214.150**	238.260	219.090
	Worst	288.690	221.070	240.270	**219.790**
	Med	286.780	**214.770**	239.355	219.130
	Mean	286.116	**215.064**	239.255	219.192
	Variance	6.5233	2.963	0.3855	**0.0339**
TOI	Best	-33.334	-34.424	-33.247	**-34.955**
	Worst	-31.274	-33.835	-32.503	**-34.641**
	Med	-32.674	-34.138	-32.943	**-34.671**
	Mean	-32.634	-34.119	-32.866	**-34.697**
	Variance	0.2931	0.0249	0.0586	**0.0065**

Table 13: Comparison of the effect of *K*, *D*, N_{femax} on SP, ONVG, and Time

MOLA		SP	ONVG	TIME (s)
D=500 N_{FEMAX}=3500	*k*=4	0.0330	**20**	2608.91
	k =10	**0.0161**	19	2610.70
	k =100	0.2786	13	2744.20
k=50 N_{FEMAX}=3500	*D*=100	0.3349	**20**	2594.60
	D=1000	0.1148	17	2589.50
	D=2000	0.1989	14	2630.45
k=50 *D*=500	N_{femax}=1000	1.2421	10	**570.74**
	N_{femax}=5000	0.2052	**20**	3831.57
	N_{femax}=10000	0.0575	13	7780.37
	N_{femax}=3500	0.0420	16	2508.90

Table 14: Performance analysis of Pareto indexes and runtime for MOLA with other proposed algorithms

	Case 1				Case 2			
	NSGA-II	MOPSO	MOIPO	MOLA	NSGA-II	MOPSO	MOIPO	MOLA
SP	70.107	605	3.247	8.024	1.506	39.353	8.978	0.034
ONVG	20	7	16	20	20	8	15	16
TIME (s)	1248	1436	1456	1188.86	2528	2472	2419	2508.9

Conclusion

In this paper, an intelligent design simulation tool was proposed to optimize CMOS op-amps. It was inferred from the results that the MOLA is suitable for creating a good trade-off between cost functions. In addition, it establishes the qualitative and quantitative characteristics of the circuits. The final evaluation indicated that MOLA is superior to other competitors. This method can be developed for optimal design domain of other analog and digital integrated circuits in the future. Also, the proposed method can be improved for designing more complex circuits with intense design specifications.

Author Contributions

Program simulation, results analysis, data collection and conclusion have been done by N. Sayyadi Shahraki under the supervision and guidance of Prof. S. H. Zahiri.

Acknowledgments

Authors would like to thank Dr. Ali Mohammadi for his help.

Conflict of Interest

The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy have been completely observed by the authors.

Abbreviations

$\omega(t)$	Inertia weight
$pbest_i^d$	Personal best position particle *i* in iteration *t*
$gbest^d$	Global best position *i* in the *d*-th dimension
C_1	Cognitive parameter
C_2	Social parameter
$U(.)$	The unit step function
$\phi_{i,j}^d$	Angle between the *i*-th ball and *j*-th ball in *d*-th dimension
N_{femax}	Maximum iteration
p_1	Probability of selection the left path or the right path
p_2	Probability of choosing a cell between the *k* cells located on the path determines
$V(x_i)$	Cell value
η	Step length
dm_i	The ratio of the drain-source voltage (V_{DS}) to the saturated drain-source voltage ($V_{DS_{sat}}$)
OSP	Output swing positive
OSN	Output swing negative
TOI	Total optimality index
A	Total area
P	Power consumption
C	Design specifications (constraints)
C_B	Specifications boundary value in

	the problem of designing an amplifier circuit
ONVG	Number of optimally non-dominated responses
SP	Spacing metric

References

[1] M. Shakibmehr, M. Lotfizad, "Design of an S-band ultra-low-noise amplifier with frequency band switching capability," Journal of Electrical and Computer Engineering Innovations, 5(1): 13–18, 2017.

[2] P. Amiri, M. Kohestani, M. Seifouri, "THD analysis in closed-loop analog PWM class-D amplifiers," Journal of Electrical and Computer Engineering Innovations, 2(1): 1–5, 2014.

[3] B. Shojaei Tabatabaei, P. Amiri, "UWB mixer improvement with regulated voltage source, " Journal of Electrical and Engineering Innovations (JECEI), 2(2): 93–99, 2014.

[4] M. Basravi, Z. H. Firouzeh, M. Maddahali, "Design of a single-layer circuit analog absorber using double-circular-loop array via the equivalent circuit model," Journal of Electrical and Computer Engineering Innovations, 5(2): 171–178, 2017.

[5] B. Liu, G. Gielen, F. V. Fernández, Automated Design of Analog and High-frequency Circuits, A Computational Intelligence Approach, Springer, Berlin, Heidelberg, 2014.

[6] S. Roostaee, H. R. Ghaffary, "Diagnosis of heart disease based on meta heuristic algorithms and clustering methods," Journal of Electrical and Computer Engineering Innovations, 2(2): 93–99, 2014.

[7] M. Ranjkesh, E. FallahCholabi, M. Pourjafari, "Optimum design of a SRM using FEM and PSO," Journal of Electrical and Computer Engineering Innovations, 2(1): 29–35, 2014.

[8] R. Omidvar1, H. Parvin, A. Eskandari, "A clustering approach by SSPCO optimization algorithm based on chaotic initial population, " Journal of Electrical and Computer Engineering Innovations, 4(1): 31–38, 2016.

[9] N. S. Shahraki, S. H. Zahiri, "Inclined planes optimization algorithm in optimal architecture of MLP neural networks, " in proc. 3rd IEEE International Conference on Pattern Recognition and Image Analysis (IPRIA): 189-194, 2017.

[10] O. Bozorg-Haddad, M. Solgi, H. A. Loáiciga, Meta-heuristic and evolutionary algorithms for engineering optimization. John Wiley & Sons, 2017.

[11] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms, " Integration VLSI Journal, 42(2): 137–148, 2009.

[12] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," Analog Integrated Circuits and Signal Processing, 63(1): 71–82, 2010.

[13] M. Barros, J. Guilherme, N. Horta, "Analog circuits optimization based on evolutionary computation techniques, " Integration, the VLSI Journal, 43(1): 136-155, 2010.

[14] R. A. Vural, T. Yildirim, "Analog circuit sizing via swarm intelligence," AEU - International Journal of Electronics and Communications, 66(9): 732–740, 2012.

[15] B. Bachir, A. Ali, M. Abdellah, "Multi-objective optimization of an operational amplifier by the ant colony optimization algorithm," Electrical and Electronic Engineering, 2(4): 230–235, 2012.

[16] S. Mallick, R. Kar, D. Mandal, S. P. Ghoshal, "Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization, " International Journal of Machine Learning and Cybernetics, 8(1): 309–331, 2017.

[17] M. Dehbashian, M. Maymandi-Nejad, "A new hybrid algorithm for analog ICs 0optimization based on the shrinking circles technique," Integration, the VLSI Journal, 5: 148-166, 2017.

[18] M. Dehbashian, M. Maymandi-Nejad, "Co-AGSA: An efficient self-adaptive approach for constrained optimization of analog IC based on the shrinking circles technique, " Integration, the VLSI Journal, 59: 218-232, 2017.

[19] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, D. Zhou, "An efficient Bayesian optimization approach for automated optimization of analog circuits, " IEEE Transactions on Circuits and Systems I, 65(6): 1954-1967, 2018.

[20] S. Dash, D. Joshi, A. Sharma, G. Trivedi, "A hierarchy in mutation of genetic algorithm and its application to multi-objective analog/RF circuit optimization," Analog Integrated Circuits and Signal Processing, 94(1): 27-47, 2018.

[21] S. Dash, D. Joshi, G. Trivedi, "Multi-Objective analog/RF circuit sizing using an improved brain storm optimization algorithm," Memetic Computing: 1-18, 2018.

[22] A. C. Sanabria-Borbón, E. Tlelo-Cuautle, "Sizing analogue integrated circuits by integer encoding and NSGA-II," IETE Technical Review, 4602(March): 1–7, 2017.

[23] A. C. Sanabria-Borbón, E. Tlelo-Cuautle, L. G. de la Fraga, "Optimal sizing of amplifiers by evolutionary algorithms with integer encoding and GM/ID design method, " in proc. NEO 2016, Springer, Cham. 731: 263-279, 2018.

[24] N. S. Shahraki, A. Mohammadi, S. Mohammadi-Esfahrood, S. H. Zahiri, "Improving the performance of analog integrated circuits using multi-objective metaheuristic algorithms, " in proc. 5th IEEE Conference on Knowledge Based Engineering and Innovation (KBEI): 822-826, 2019.

[25] E. Afacan, "Inversion coefficient optimization based analog/RF circuit design automation," Microelectronics Journal, 83: 86-93, 2019.

[26] M. Hasanzadeh-Mofrad, A. Rezvanian, "Learning automata clustering," Journal of Computational Science, 24: 379–388, 2018.

[27] M. Ahangaran, N. Taghizadeh, H. Beigy, "Associative cellular learning automata and its applications," Applied Soft Computing, 53: 1-18, 2017.

[28] B. Damerchilu, M. S. Norouzzadeh, M. R. Meybodi, "Motion estimation using learning automata," Machine Vision and Applications, 27(7): 1047-1061, 2016.

[29] N. Kumar, J. H. Lee, J. J. Rodrigues, "Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach," IEEE Transactions on Intelligent Transportation Systems, 16(3): 1148-1161, 2015.

[30] A. L. Bazzan, "Aligning individual and collective welfare in complex socio-technical systems by combining metaheuristics and reinforcement learning engineering," Applications of Artificial Intelligence, 79: 23-33, 2019.

[31] M. Rezapoor Mirsaleh, M. R. Meybodi, "Balancing exploration and exploitation in memetic algorithms: A learning automata approach," Computational Intelligence, 34(1): 282-309, 2018.

[32] W. Li, E. Özcan, R. John, "A learning automata based multiobjective hyper-heuristic," IEEE Transactions on Evolutionary Computation, 21(1): 59–73, 2017.

[33] M. L. Tsetlin, Automaton Theory and Modeling of Biological Systems, 102 of Mathematics in Science and Engineering. Academic Press, New York, 1973.

[34] F. Hourfar, H. J. Bidgoly, B. Moshiri, K. Salahshoor, A. Elkamel, "A reinforcement learning approach for waterflooding optimization in petroleum reservoirs," Engineering Applications of Artificial Intelligence. 77: 98-116, 2019.

[35] H. L. Liao, Q. H. Wu, "Multi-objective optimization by learning automata, "Journal of Global Optimization, 55(2): 459–487, 2013.

[36] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II, " IEEE Transactions on Evolutionary Computation, 6(2): 182–197, 2002.

[37] C. A. Coello Coello, G. T. Pulido, M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," IEEE Transactions on Evolutionary computation, 8(3): 256–279, 2004.

[38] A. Mohammadi, M. Mohammadi, S. H. Zahiri, "Design of optimal CMOS ring oscillator using an intelligent optimization tool," Soft Computing, 22(4): 8151-8166, 2018.

[39] J. Kennedy, R. Eberhart, "Particle swarm optimization," in Proc. IEEE International Conference on Neural Networks, 4): 1942–1948, 1995.

[40] M. H. Mozaffari, H. Abdy, S. H. Zahiri, "IPO: An inclined planes system optimization algorithm," Computing and Informatics, 35(1): 222-240, 2016.

Biographies

Najmeh Sayyadi Shahraki received the B.Sc. degree in Electrical Engineering from Shahrekord University, in 2010 and the M.Sc. degree in Electrical Engineering from University of Birjand, in 2014. She is currently a Ph.D. candidate in the University of Birjand. Her research interests include soft computing and its applications, engineering optimization, nature-inspired optimization intelligent algorithms, artificial intelligence, and electronic circuit design. Her personal page on ORCID is available in https://orcid.org/0000-0001-6332-3702.

Seyed Hamid Zahiri received the B.Sc., M.Sc. and Ph.D. degrees in Electronics Engineering respectively from Sharif University of Technology, Tehran, Tarbiat Modarres University, Tehran, and Mashhad Ferdowsi University, Mashhad, Iran, in 1993, 1995, and 2005. Currently, he is Professor with the Department of Electronics Engineering, University of Birjand, Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms, swarm intelligence algorithms, and soft computing. His personal page on ORCID is available in: https://orcid.org/0000-0002-1280-8133.

How to cite this paper:
N. Sayyadi Shahraki, S. H. Zahiri, “Low-area/low-power CMOS op-amps design based on total optimality index using reinforcement learning approach” Journal of Electrical and Computer Engineering Innovations, 6(2): 199-214, 2018.
DOI: 10.22061/JECEI.2019.5212.203
URL: http://jecei.srttu.edu/journal/authors.note