IUST Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper

Multi-Objective Learning Automata for Design and Optimization a Two-Stage CMOS Operational Amplifier

N. Sayyadi Shahraki* and S. H. Zahiri*(C.A.)

Abstract: In this paper, we propose an efficient approach to design optimization of analog circuits that is based on the reinforcement learning method. In this work, Multi-Objective Learning Automata (MOLA) is used to design a two-stage CMOS operational amplifier (op-amp) in 0.25μm technology. The aim is optimizing power consumption and area so as to achieve minimum Total Optimality Index (TOI), as a new and comprehensive proposed criterion, and also meet different design specifications such as DC gain, Gain-Band Width product (GBW), Phase Margin (PM), Slew Rate (SR), Common Mode Rejection Ratio (CMRR), Power Supply Rejection Ratio (PSRR), etc. The proposed MOLA contains several automata and each automaton is responsible for searching one dimension. The workability of the proposed approach is evaluated in comparison with the most well-known category of intelligent meta-heuristic Multi-Objective Optimization (MOO) methods such as Particle Swarm Optimization (PSO), Inclined Planes system Optimization (IPO), Gray Wolf Optimization (GWO) and Non-dominated Sorting Genetic Algorithm II (NSGA-II). The performance of the proposed MOLA is demonstrated in finding optimal Pareto fronts with two criteria Overall Non-dominated Vector Generation (ONVG) and Spacing (SP). In simulations, for the desired application, it has been shown through Computer-Aided Design (CAD) tool that MOLA-based solutions produce better results.

Keywords: Analog Circuit Design, Area and Power Optimization, Multi-Objective Learning Automata, Total Optimality Index.

1 Introduction

THE main field of this paper is related to three topics: integrated circuit design, meta-heuristic optimization methods, and the use of Learning Automata (LA) based on the reinforcement learning approach. Its main topic is the relationship between the Multi-Objective Learning Automata (MOLA) in terms of optimal design of operational amplifiers (op-amps), which are one of the most used modules in analog integrated circuits. In the following, in three different parts, these main topics are described separately.

Op-amps are one of the most important sub-sections in analog circuits. A two-stage op-amp is used widely for various applications due to its robustness and structure. For example in [1], a novel low-voltage two-stage operational amplifier employing resistive biasing is presented. In [1], for each stage, an independent common-mode feedback a circuit has been used which reduced the power consumption and increased output voltage swing. Analog circuit design is a challenging process which involves the characterization of complex trade-offs between nonlinear objectives and the specifications such as DC gain, Gain-Band Width product (GBW), Phase Margin (PM), Slew Rate (SR), Common Mode Rejection Ratio (CMRR), Power Supply Rejection Ratio (PSRR), etc. Due to the complexity of analog circuits, their manual design with high performance and low power is not simple. Therefore, intelligent optimization methods are required for automation and optimal sizing of CMOS analog ICs design [2]. One of the most well-known categories is meta-heuristic algorithms.

Iranian Journal of Electrical and Electronic Engineering, 2020.
Paper first received 11 April 2019, revised 14 June 2019, and accepted 28 June 2019.
* The authors are with the Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.
E-mails: sayyadi.najmeh@birjand.ac.ir and hzahiri@birjand.ac.ir.
Corresponding Author: S. H. Zahiri.

Meta-heuristic algorithms have high performance and the ability for solving optimization problems. The purpose of meta-heuristic algorithms is to find proper values for the decision design parameters of an optimization problem to optimize one/multiple objective function [3]. These 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 perform circuit design in the form of the Single-Objective (SO) and Multi-Objective (MO) optimization. Usually, analog circuits have several conflicting performances. For this reason, the Multi-Objective Optimization (MOO) has also been introduced for the automated design of CMOS analog ICs. It would be useful to produce a set of results for the designers with the best trade-off between performances. Unlike SO optimization methods, a MOO algorithm attempts to find non-dominated solutions during the optimization process. In designing amplifier circuits, power and area conflict with each other; so that by decreasing the channel length, the speed of MOSFETs increases (which means reducing the delay). This increase in speed leads to increased power consumption. Therefore, MO techniques are used simultaneously to reduce the power consumption and area of MOSFETs [4]. Meta-heuristic methods are applied for MO analog circuit optimizations.

One of the important MO methods is the NSGA-II evolutionary algorithm. NSGA-II was proposed in 2002 by Deb [5]. It is a modified version of the Genetic Algorithm (GA) [6] with the elitist approach. The GA concept is developed from natural evolution process. Based on the Darwin theory "survival of fittest", the GA mimics the natural evolution method. The elitism approach used to copy best parents and offspring (i.e., child) produce by the genetic operators. In NSGA-II algorithm, non-dominated solution is obtained from the current parents and their offspring using objective functions. This algorithm has shown its ability in many applications. Therefore, in this paper, it is used as one of the competing algorithms and is assigned in a sub-section [5].

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. For example, GSA-PSO algorithm was used to optimization differential amplifier circuit with current mirror load and CMOS two-stage operational amplifier circuit [7]. In [8], a new approach is proposed to automatically size three conventional amplifier circuits. In order to enhance the performance of automatic sizing of analog circuits, a new shrinking circles technique has been used [9]. A Weighted Expected Improvement based Bayesian Optimization (WEIBO) is proposed for the automated analog circuit sizing [10]. The hierarchical Non-dominated Sorting Genetic Algorithm II (hNSGA–II) [11] and Improved Brain Storm Optimization (IMBSO) [12] algorithms are proposed for MOO of circuits. An Inversion Coefficient (IC) optimization-based analog/RF circuit sizing approach is proposed in three different circuits [13]. One of the other important approaches that is ignored in the optimal design of analog circuits and can be applied along with meta-heuristic algorithms is LA-based on reinforcement learning.

LA is a reinforcement learning approach that is an unsupervised optimization method and 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 in, for instance data mining [14,15], image processing [16,17], and optimization [18-20]. The general technique of choosing an action from a series of actions is related to the highest reward compared to other actions. This result is achieved through interactions with the environment in terms of a sequence of repetitive feedback cycles. By learning to choose the optimal action, the automata adapt themselves to the environment, needless to have detailed information about the environment model [21]. The idea of LA was first introduced by Tsetlin to model biological learning mechanism [22]. In LA research, various types of LA-based algorithms have been developed. In this work, we have used the MO version of Learning Automata (MOLA) method [23] for the automated design of a two-stage CMOS op-amp. This paper focuses on the design of circuit parameters, considering the assumption of the appropriate topology is selected by the designer.

This paper contains several contributions that are listed as follows:

- A new application of LA for MOO in the optimal design of CMOS analog IC.
- Proper definition of design parameters and objective functions to create an effective trade-off between performance characteristics.
- Implementation of an automated design simulation tool by creating a link between two usable software environments.
- Providing a comprehensive criterion to evaluate the proposed approach due to the simultaneous effect of objectives and design specifications on the optimization problem.
- The statistical evaluation of the proposed approach based on numerical results obtained from circuit simulations with other competing algorithms.

This paper is organized as follows. Section 2 introduces our proposed tool, case study, and along with a description of the MOLA method and rival meta-heuristic MOO algorithms. In Section 3, the considerations for design and optimization of the proposed circuit are provided. The simulation results are reported in Section 4. Finally, in Section 5 the conclusion is expressed.

2 Meta-heuristic Approaches for Multi-Objective Simulation-based Optimization

In real applications, we constantly deal with problems that under specific circumstances are faced with several objective 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 objective functions. These objectives are usually in trade-off. So, the meta-heuristic approaches are the best candidate for solving them. 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 with each objective function with an acceptable level, is defined as optimal solutions.

In this section, an automated MO simulation-based optimization approach is proposed for intelligent and optimal design of analog IC. The proposed Computer-Aided Design (CAD) tool is applied for this purpose. It should be noted that analog circuits are simulated by the HSPICE simulator. 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 current.

Continue on this section, the desired amplifier circuit, the MOLA method with other MOO algorithms employed is explained.

2.1 Two-stage CMOS Op-Amp

In order to show the performance of the proposed MOLA method in the design of analog circuits, a two-stage CMOS op-amp in 0.25µm technology is used. There are 13 design parameters in this circuit. In Fig. 2, a two-stage CMOS op-amp is shown with Miller compensation capacitance. Miller's compensation technique is used to frequency compensation in this amplifier to utilize bandwidth, phase margin, and circuit stability. This movement of the amplifier pole to reduce the frequency of dominant pole improves the amplifier stability. Therefore, a low-frequency pole can be established with moderate capacitor value, saving considerable chip area [24]. 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 $M_1 \equiv M_2$, $M_3 \equiv M_4$, and $M_5 \equiv M_8$. Furthermore, the positive power supply (V_{DD}) and the negative power supply (V_{SS}) are equal to 2.5V and -2.5V, respectively [8].This circuit set values for the C_C and C_L that provide $C_C > 0.22C_L$ [7]. Desired specifications (small-signal differential voltage gain (DC gain), Gain-Band Width product (GBW), Common Mode Rejection Ratio (CMRR), Power Supply Rejection Ratio (PSRR), etc) are in accordance with Table 1.

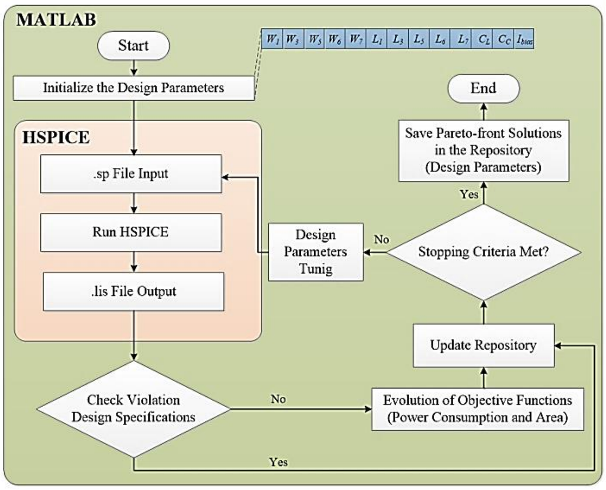

Fig. 1 The general structure of the automated circuit design simulation tool based on the intelligent MOO.

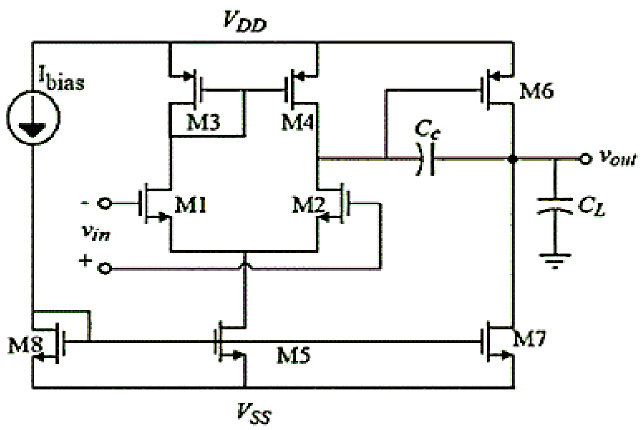

Fig. 2 The proposed two-stage CMOS op-amp circuit [8].

Table 1 Desired characteristics of two-stage CMOS OP-AMP.

Design specifications	Constraints
DC gain [dB]	≥70
GBW [MHz]	≥2
Phase Margin [deg]	≥50
Slew Rate [V/µs]	≥1.5
Output Swing [V]	≥2
CMRR [dB]	≥70
$PSRR^+$ [dB]	≥70
$PSRR^-$ [dB]	≥70
$M_1, \ldots, M_8$	Saturation

In this paper, for the first time, the MOLA method is used along with four rival MOO algorithms (called NSGA-II [5], MOPSO [25], MOIPO [26], and MOGWO [27]). In the following, the description of the proposed algorithm is presented with four competing algorithms.

2.2 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 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 offsprings are combined before partitioning the new combined pool into fronts [5]. The flowchart of the NSGA-II algorithm is depicted in Fig. 3.

2.3 Multi-Objective Particle Swarm Optimization (MOPSO)

PSO is one of the most important intelligent optimization algorithms [28]. One of the most popular and effective proposals for MO versions of the PSO optimization algorithm is presented in [25]. The position of the non-dominated particles is stored in a repository. Then, the search space is divided into some hypercubes. These non-dominated particles are located in accordance with the values of their objective functions in the hypercubes. While the maximum number of iterations is not provided, the speed and position of the particles are updated. Then the contents of the repository are updated. This update consists the inserting all the currently non-dominated locations in the repository and the removal of the dominated locations from it during the process. Since the repository size is limited, whenever it gets full, hypercubes that contain more particles in themselves are identified and the excess particles are randomly removed from the hypercubes [25]. The flowchart of the MOPSO algorithm is shown in Fig. 4.

2.4 Multi-Objective Inclined Planes system Optimization (MOIPO)

The search factors in the 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 [29]. The MO version of the algorithm has been created in [26]. Also, Fig. 5 shows its flowchart.

2.5 Multi-Objective Gray Wolf Optimization (MOGWO)

The Gray Wolf Optimization algorithm is inspired by the hierarchical structure of the wolf position in the group as well as its structure and duties in hunting. In this algorithm, the search factors corresponding to wolves, the hunting process corresponds to the process of finding the optimal response and the location of the hunt corresponding to the optimal response position [30]. MOGWO flowchart is shown in Fig. 6.

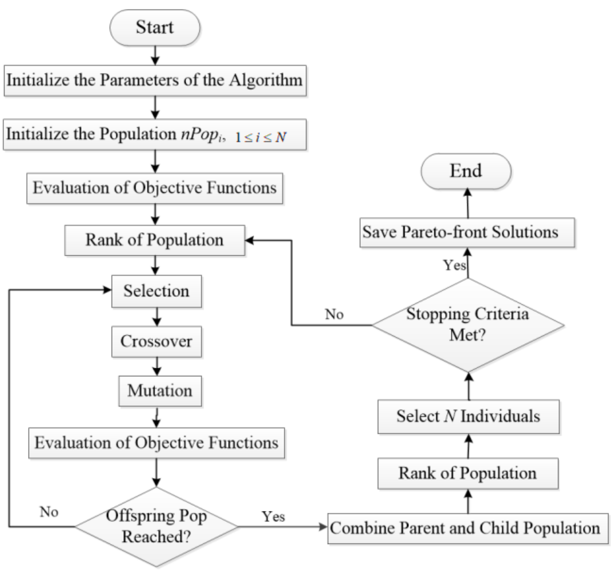

Fig. 3 Flowchart of the NSGA-II algorithm.

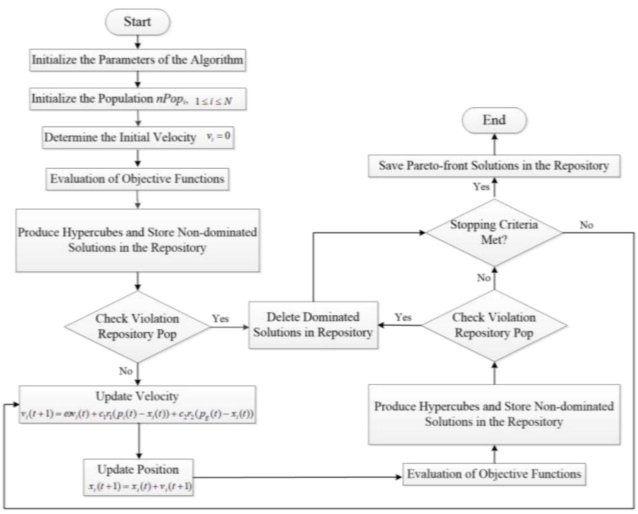

Fig. 4 Flowchart of the MOPSO algorithm.

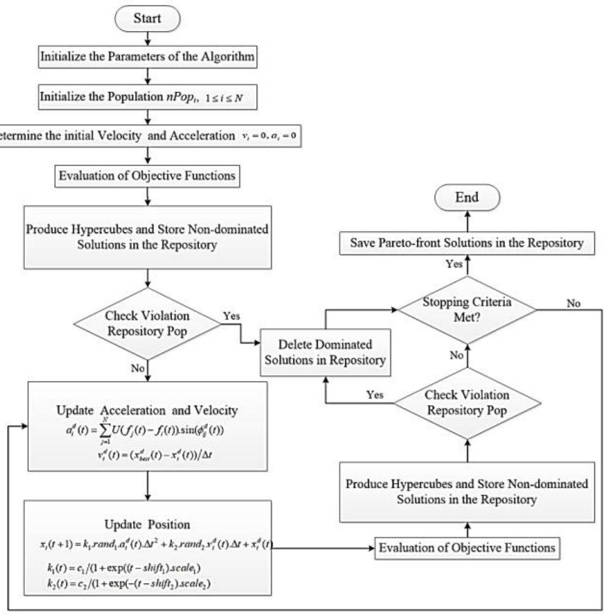

Fig. 5 Flowchart of the MOIPO algorithm.

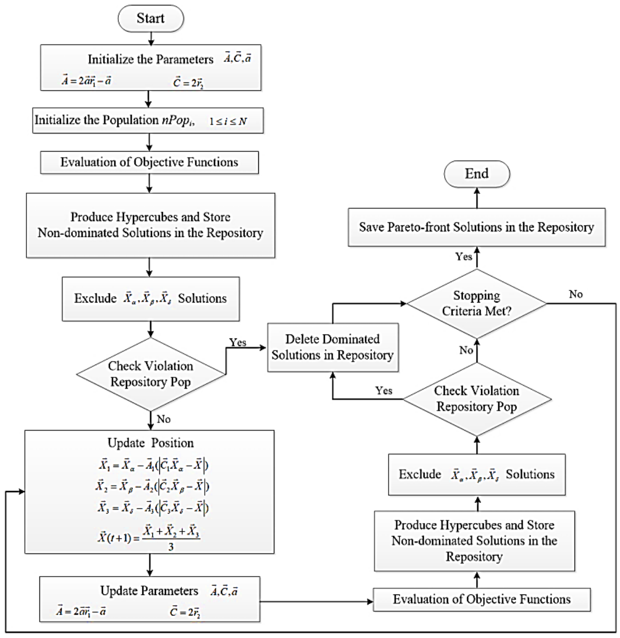

Fig. 6 Flowchart of the MOGWO algorithm.

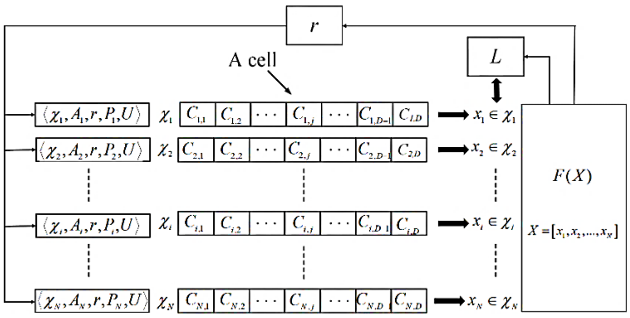

Fig. 7 The structure of learning automata for MOLA [23].

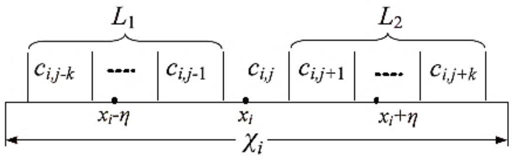

Fig. 8 The two possible paths taken by a search starting at dimensional state xi on the *i*-th dimension [23].

2.6 Multi-Objective 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 method is equal to the dimensions of the problem. For *N*-dimensional problem, the MOLA includes *N* automata [23]. The structure of learning automata for MOLA has been shown in Fig. 7. Each automaton is responsible for searching one dimension and acts independently in the environment.

The *i*-th learning automata is defined by $<x_i, A_i, r, P_i, U>$, where $\chi_i = \{x_i\}$ is the set of possible states on the *i*-th dimension. Also, x_i is the dimensional state on the *i*-th dimension ($x_i \in [x_{\min,i}, x_{\max,i}]$), the minimum and maximum values in the *i*-th dimension are $x_{min,i}$ and $x_{max,i}$, respectively. In MOLA, $A_i = \{a_{l,\eta}\}$ is the set of possible actions which the learning automata can take on dimension *i*, $a_{l,\eta}$ indicates that an action moves left ($l = 1$) or right ($l = 2$) and η is step length. Note that *r* is a scalar value and shows reinforcement signal. It produced through the environment to indicate the quality of the action of moving x_i in a step length on the selected path. Also, P_i consists of two probabilities p_1 and p_2. Where p_1 shows the probability of selecting the left path or the right path on *i*-th dimension. Assume that the right path is selected, the probability of choosing a cell between the *k* cells located on the path determines by the probability of p_2. Also, *U* is a scheme adopted to calculate the probabilities of actions, *P*.

In the MOLA method, each dimension is divided into *D* cells. This means that χ_i is divided into *D* subsets and subset includes all dimensional states located in the cell. Therefore, $N\times D$ cells are produced for an *N*-dimensional search space. Considering the $x_{\min,i}$ and $x_{\max,i}$ are minimum and maximum values in the *i*-th dimension, respectively. Also, *D* is the number of divisions of each cell. Then, $\omega_{c,i}$ is the width of a cell in *i*-th dimension, and it is calculated by (1).

$$\omega_{c,i} = \frac{x_{\max,i} - x_{\min,i}}{D} \tag{1}$$

In the beginning of the action search, in order to estimate the choice of a better solution on the path, we should be able to choose one of two possible directions. In other words, the path values must be determined by the cell values on the path.

As shown in Fig. 8, the value of $L_2(x_i)$ is specified by the values of k adjacent cells on the right path, where k is the integer predefined value and $c_{i,j}$ is j-th cell in i-th dimension. Also, j is calculated by (2).

$$j = \text{floor}\left(\frac{x_i - x_{\min,i}}{\omega_{c,i}}\right) \tag{2}$$

The value of a path can be estimated as (3). Where $v^*_{i,m}$ presents the m-th element of the vector which is placed on path l. Also, λ_1 is calculated with $0 \le \lambda_1 \le 1$ and $(1-\lambda_1)\sum_{m=1}^{k-1}\lambda_1^{m-1} + \lambda_1^{k-1} = 1$, subject to $(1-\lambda_1)\lambda_1^{k-2} \ge \lambda_1^{k-1}$.

$$L_l(x_i) = (1-\lambda_1)\sum_{m=1}^{k-1}\lambda_1^{m-1}v^*_{l,m} + \lambda_1^{k-1}v^*_{l,k} \qquad \forall l = 1,2 \tag{3}$$

Two probabilities of p_1 and p_2 are obtained from (4) and (5). Where $V(x_i)$ is cell value. Temperature τ creates a trade-off between exploration and exploitation.

$$p_1\left(L_l(x_i)\right) = \frac{e^{\frac{L_l(x_i)}{\tau}}}{\sum_{s=1}^{2} e^{\frac{L_s(x_i)}{\tau}}} \qquad \forall l = 1,2 \tag{4}$$

$$p_2(c_{i,j+s}) = \frac{e^{\frac{V(x_i)\big|_{x_i \in c_{i,j+s}}}{2\tau}}}{\sum_{z=1}^{k} e^{\frac{V(x_i)\big|_{x_i \in c_{i,j+z}}}{2\tau}}} \qquad \forall l = 1,2\ ,\ s = 1,\ldots,k \tag{5}$$

By choosing a cell, an action moves to the new cell with a step length that can be denoted as η. Which is calculated in accordance with (6). In (6), the distance (in the form of the number of cells) between the current cell and the selected cell is ξ and ζ is a random number ($\zeta \in (0, 1]$).

$$\eta = (\xi + \zeta)\,\omega_{c,i} \tag{6}$$

Therefore, 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 $x_i = x_i + \eta$. Then a reinforcement signal is used to check the new dimensional state x_i. When dimensional state x_i moves to x'_i, the i-th element of the current state $X(x_i)$ is replaced by $X(x'_i)$. Reinforcement signal is assigned to cell $c_{i,j}$ according to (7). In (7), $r = 1$ indicates that the solution is desirable and $r = 0$ presents an undesirable response.

$$r\left(X(x'_i)\right) = \begin{cases} 1, & \text{if } X(x'_i) \text{ is a non-dominated solution} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The reinforcement signal is applied to update the cell value of cell $c_{i,j}$ which dimensional state x'_i. Considering that $L_{\max}(x_i) = \max\{L_1(x_i), L_1(x_i)\}$ and $L_{\min}(x_i) = \min\{L_1(x_i), L_1(x_i)\}$ are the two estimated path values at x_i.Also, weights α_1 and $(1-\alpha_1)$ present the influence of previous estimates and path values on the new estimate, respectively. Then, the value of cell $c_{i,j}$, where the current dimensional state x_i locates, is updated as (8). In (8), the $L_{\max}(x_i)$ has a greater influence on the cell value than $L_{\min}(x_i)$, therefore parameter λ_2 should be given such that $((1-\lambda_2) > \lambda_2)$.

$$\begin{aligned} V(x_i)\big|_{x_i \in c_{i,j}} \leftarrow\ & r\left(X(x_i)\right) + \alpha_1 V(x_i)\big|_{x_i \in c_{i,j}} \\ & + (1-\alpha_1)\left((1-\lambda_2)L_{\max}(x_i) + \lambda_2 L_{\min}(x_i)\right) \end{aligned} \tag{8}$$

A repository saves all non-dominated solutions in an elite list, L. If $X(x'_i)$ dominates all of the L solutions, it is known as X_{best} and then L is updated. In (9), the relation between X and X_{best} is shown.

$$X_{best} \leftarrow \begin{cases} X(x'_i), & \text{if } X(x'_i) \text{ is a non-dominated solution} \\ X_{best}, & \text{otherwise} \end{cases} \tag{9}$$

where

$$X(x'_i) = \left[x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_N\right] \tag{10}$$

Then L is updated according to (11). Where B is set of the solutions which is dominated by X_{best}

$$L \leftarrow \begin{cases} L \cup \{X_{best}\} - B, & \text{if } r = 1 \\ L, & \text{otherwise} \end{cases} \tag{11}$$

To increase the variety and explore the solutions of radiation solutions, we apply perturbations according to (12). Where β is a random variable ($\beta \in [0, 1]$).

$$X' \leftarrow X + \Delta + \beta(X_{best} - X) \tag{12}$$

Also, Δ is calculated according to (13). Where ζ is a random variable ($\zeta \in [0, k/D]$). The input to the sign function is the subtraction of the two adjacent cell values of $c_{i,j}$, which is represented by (14)

$$\Delta_i = \text{sign}(\kappa)\zeta(x_{\max,i} - x_{\min,i}) \tag{13}$$

$$\kappa = V(y_i)\big|_{y_i \in c_{i,j+1}} - V(z_i)\big|_{z_i \in c_{i,j-1}} \tag{14}$$

The sign function acts as (15)

$$\text{sign}(\kappa) = \begin{cases} 1, & \kappa > 0 \\ -1, & \kappa < 0 \end{cases} \tag{15}$$

The N_{femax} is a given maximum number of objective functions evaluations by which the MOLA computations proceed in episodes.

3 The Considerations for Design and Optimization of the Proposed Circuit

The approach is based on intelligent sizing to power and area optimization using the proposed meta-heuristic methods. So, in this Section, the objective functions, the proposed new index, and Pareto-front evaluation criteria are presented.

3.1 Objective Functions

In this paper, intelligent methods are used to optimize the two important and essential indicators of amplifier circuits that are in conflict with each other, namely power consumption and area. These objective functions are considered as follows:

- Minimizing power consumption,
- Minimizing the area.

3.2 Total Optimality Index (TOI)

In this paper, due to the diversity and multiplicity of qualitative indicators in the design problem, a total criterion is presented that illustrates the success of the optimization method. This criterion can be used to investigate the performance of the proposed optimization method in the design problem. Therefore, a criterion called Total Optimality Index (TOI) is proposed. The TOI has been introduced to express the impact of the design specifications and the objective functions of the problem. The lower value of TOI represents the more favorable response. In the paper, TOI is not considered as an objective function.

The main purpose of the proposed index is to provide a comprehensive criterion for verifying the superiority and success of an intelligent optimization method employed in the optimal design of the problem; so that the audience, through the numerical values of this index, can grasp the definitive and comprehensive success of the proposed approach in this paper. Therefore, its scientific basis is based on the merging of the parameters the objective functions, problem constraints, and some mathematical tools in such a way as to achieve the optimal value of each of the parameters can be found in the minimum/maximum value of the index. For this purpose, in addition to incorporating the optimal values of the objective functions, the circuit constraints are also intelligently taken into account. The resulting values of this index are to be minimized, and its low value represents the success of an optimization method to overcome the design challenge and to achieve global optimal solutions while satisfying the exact constraints of the problem.

Assuming that A is the total area of the MOSFET in μm^2 and P is the power consumption in mW (as the objective functions), C is design specifications (constraints) and C_B is specifications boundary value in the problem of designing an amplifier circuit. Then, the index is defined as follows:

$$TOI = \frac{\text{normalized}\left(A[\mu m^2].P[mW]\right)}{\dfrac{\text{sum}\left(|C|\right)}{\text{sum}\left(|C_B|\right)}} \tag{16}$$

In (16), 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 $|C_B|$ is used. The most desirable TOI (minimum) is created by minimizing objective functions and maximizing the design specifications. The design of the TOI is such that the main focus is on the objective functions of the problem and a minor improvement in one of them will minimize the TOI.

3.3 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. Where $|PF_{known}|$ is the number of vectors in PF_{known} (known/current Pareto-front).

$$ONVG \equiv |PF_{known}| \tag{17}$$

- **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 (18)). Where $d_i = \min_j(|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$, i, j=1,…, n, $\bar{d}$ is the mean of all d_i, and n is the number of vectors in PF_{known} ($|PF_{known}|$). So that, SP = 0, means that all members are spaced evenly apart in [26].

$$SP = \sqrt{\frac{\sum_{i=1}^{n}(\bar{d} - d_i)^2}{(n-1)}} \tag{18}$$

4 Simulation Results

In this section, the results and analysis are presented in the optimization of two-stage CMOS op-amp. All the results are reported in the form of the values of fitness objective functions, design parameters, design specifications, and TOI. The best, worst, mean, and variance of the values of the objective functions and the TOI are presented for proposed methods in the best run. Figures of the Pareto-front and the design specifications of the two-stage CMOS op-amp including DC gain, Phase Margins, PSRR, and Slew Rate are plotted by the proposed methods. In addition, the Pareto indexes and the runtime of MOLA performance are analyzed in comparison with other proposed algorithms for 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 vector of design parameters that should be determined by the proposed methods is as follows:

$$X = [W_1, W_3, W_5, W_6, W_7, L_1, L_3, L_5, L_7, C_L, C_C, I_{bias}] \quad (19)$$

The details on design parameters for two-stage CMOS op-amp are listed in Table 2. Also, in Table 3 all control parameters of the proposed methods in this paper are presented.

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.

Tables 4-9 show the best run of the algorithms for this circuit that is generated by the best TOI. In all tables, the bolded responses show the best values in terms of design specifications, objective functions, and TOI in the best run. A solution marked by a sub-line expresses a solution in the desired Pareto-front, which has the best TOI (minimum). MOLA method is able to produce the minimum area and power consumption with the values 72.825μm^2 and 0.560mW, respectively. 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. The algorithms performance presents an intelligent optimization and trade-off between objectives of the problem. The variety and the number of presented Pareto-front solutions provide a wide range of selection for the circuit designer. According to tables, this superiority, relative to all the Pareto-optimal solutions of algorithms, is achieved with 36.36, 18.18, 18.18, and 9.09% by MOLA, MOGWO, MOPSO, and NSGA-II, respectively. Due to the well-known and widely used of NSGA-II, it was expected to perform better than other algorithms. Although it has not been able to

Table 2 The range of design parameters.

Design parameters	Lower bound	Upper bound
W [μm]	5	40
L [μm]	0.25	2
I_{bias} [μA]	20	40
C_C [pF]	2	20
C_L [pF]	7	15

Table 3 Control settings.

Parameters	NSGA-II	MOPSO	MOIPO	MOGWO	MOLA
Total Run	20	20	20	20	20
MaxIt / N_{femax}	100	100	100	100	2000
nPop	20	20	20	20	1
nRep	20	20	20	20	20
nGrid	4	4	4	4	4
α	0.1	0.1	0.1	0.1	0.1
β	4	4	4	4	4
γ	2	2	2	2	2
P_c	0.9	—	—	—	—
P_m	0.1	—	—	—	—
η_c	2	—	—	—	—
η_m	18	—	—	—	—
C_1 / $c1$	—	1.4962	0.1	—	—
C_2 / $c2$	—	1.4962	3.05	—	—
w	—	1	—	—	—
w_{damp}	—	0.73	—	—	—
shift1	—	—	100	—	—
shift2	—	—	300	—	—
scale1	—	—	0.03	—	—
scale2	—	—	0.03	—	—
$\vec{a}$	—	—	—	∈ [0, 2]	—
$\vec{r}_1$	—	—	—	∈ [0,1]	—
$\vec{r}_2$	—	—	—	∈ [0,1]	—
D	—	—	—	—	500
k	—	—	—	—	50
α_1	—	—	—	—	∈ [0, 1]
λ_1	—	—	—	—	0.5
λ_2	—	—	—	—	∈ [0, 1]
τ	—	—	—	—	∈ [0, 0.5]

demonstrate its superiority to others, especially the proposed method MOLA; but it has in many cases been able to provide good results than MOPSO, MOIPO, and MOGWO.

Table 4 Optimal design of parameters, specifications, objectives, and TOI for MOLA method.

MOLA		Pareto-solutions								
		1	2	3	4	…	17	18	19	20
Design Parameters	$W_1/L_1=W_2/L_2$ [μm/μm]	5.104/1.654	5.104/1.654	5.104/1.654	5.104/1.654	…	5.104/1.654	5.104/1.654	5.104/1.654	5.104/1.654
	$W_3/L_3=W_4/L_4$ [μm/μm]	6.813/0.537	6.813/0.537	6.813/0.537	6.813/0.537	…	6.813/0.537	6.813/0.537	6.813/0.537	6.813/0.537
	$W_5/L_5=W_8/L_8$ [μm/μm]	9.221/1.479	9.221/1.479	9.221/1.479	9.221/1.479	…	9.221/1.479	9.221/1.479	9.221/1.479	9.221/1.479
	W_6/L_6 [μm/μm]	25.988/0.735	25.988/0.481	25.988/0.735	25.988/0.580	…	25.988/0.481	25.988/0.481	25.988/0.481	25.988/0.481
	W_7/L_7 [μm/μm]	12.071/0.797	12.071/0.935	12.071/0.91	12.071/0.91	…	12.071/0.888	12.071/0.882	12.071/0.802	12.071/0.935
	C_C [pF]	7.831	7.831	7.831	7.831	…	7.831	7.831	7.831	7.831
	C_L [pF]	8.868	8.891	8.868	8.868	…	8.868	8.868	8.868	8.868
	I_{bias} [μA]	20.037	20.037	20.037	20.037	…	20.037	20.037	20.037	20.037
Design Specifications	DC gain [dB]	72.751	72.104	73.808	73.546	…	71.648	71.925	72.098	72.104
	GBW [MHz]	2.110	2.125	2.113	2.124	…	2.127	2.127	2.127	2.126
	Phase margin [deg]	50.596	59.362	50.244	55.904	…	60.616	60.247	59.756	59.394
	Slew rate [V/μs]	3.017	3.020	3.021	3.023	…	3.016	3.018	3.020	3.022
	Output swing [V]	2.306	2.334	2.309	2.331	…	2.341	2.337	2.343	2.334
	CMRR [dB[	104.156	78.64	103.110	79.807	…	79.363	79.178	78.896	78.644
	$PSRR^+$[dB]	82.222	82.460	82.530	80.819	…	78.621	78.501	78.613	78.690
	$PSRR^-$ [dB]	82.892	85.131	85.797	89.585	…	92.825	88.981	92.522	95.485
Objectives	Area [μm^2]	80.199	75.263	81.563	77.535	…	72.825	73.658	74.624	75.264
	Power consumption [mW]	0.562	0.631	0.560	0.604	…	0.650	0.645	0.637	0.631
TOI		**0.0526**	0.0560	0.0530	0.0562	…	0.0567	0.0566	0.0563	0.0560

Table 5 Optimal design of parameters, specifications, objectives, and TOI for MOGWO algorithm.

MOGWO		Pareto-solutions								
		1	2	3	4	…	8	9	10	11
Design Parameters	$W_1/L_1=W_2/L_2$ [μm/μm]	6.689/1.62	5.192/1.159	5.175/1.207	5.245/1.158	…	5.104/1.064	5.242/1.144	5.16/1.115	5.208/1.123
	$W_3/L_3=W_4/L_4$ [μm/μm]	27.493/1.618	21.035/1.122	15.907/0.96	18.37/1.068	…	16.004/0.947	18.21/1.017	16.677/0.989	17.01/0.984
	$W_5/L_5=W_8/L_8$ [μm/μm]	19.377/1.482	13.251/0.972	10.114/0.84	11.801/0.937	…	10.606/0.813	11.716/0.941	10.73/0.876	11.055/0.889
	W_6/L_6 [μm/μm]	32.483/0.311	24.632/0.302	22.511/0.295	24.377/0.307	…	23.732/0.286	24.942/0.305	23.523/0.294	24.103/0.296
	W_7/L_7 [μm/μm]	30.656/1.824	20.274/1.283	17.765/1.183	18.52/1.251	…	19.031/1.086	18.83/1.263	18.322/1.18	18.688/1.186
	C_C [pF]	5.397	5.014	4.035	4.593	…	4.227	4.398	4.233	4.283
	C_L [pF]	10.163	7.785	7.12	7.571	…	7	7.496	7.092	7.21
	I_{bias} [μA]	20	20.035	20	20	…	20	20.107	20	20.036
Design Specifications	DC gain [dB]	**74.121**	72.214	71.014	72.049	…	70.21	71.836	70.972	71.127
	GBW [MHz]	3.311	3.844	4.661	4.177	…	**4.722**	4.386	4.601	4.552
	Phase margin [deg]	50.274	51.086	51.864	50.205	…	**53.962**	50.214	51.982	51.876
	Slew rate [V/μs]	4.095	4.568	**5.725**	4.984	…	5.503	5.227	5.439	5.379
	Output swing [V]	**2.355**	2.353	2.343	2.348	…	2.350	2.347	2.346	2.347
	CMRR [dB]	**80.3413**	78.2213	77.1162	77.7234	…	76.6052	77.7917	77.1882	77.3476
	$PSRR^+$[dB]	**77.867**	75.299	74.342	75.544	…	73.762	75.219	74.459	74.565
	$PSRR^-$ [dB]	**92.281**	92.281	86.285	87.811	…	85.978	87.907	86.858	87.18
Objectives	Area [μm^2]	234.0919	118.4479	87.68215	104.1531	…	**85.87326**	102.4719	91.82859	94.12709
	Power consumption [mW]	**0.573**	0.588	0.641	0.598	…	0.674	0.607	0.636	0.633
TOI		0.1778	0.0902	**0.0710**	0.0800	…	0.0730	0.0797	0.0742	0.0758

Table 6 Optimal design of parameters, specifications, objectives, and TOI for MOIPO algorithm.

MOIPO		Pareto-solutions								
		1	2	3	4	…	13	14	15	16
Design Parameters	$W_1/L_1=W_2/L_2$ [μm/μm]	9.525/1.263	9.216/1.227	9.701/1.279	9.791/1.28	…	9.845/1.289	14.047/1.321	9.943/1.299	9.999/1.306
	$W_3/L_3=W_4/L_4$ [μm/μm]	24.843/0.77	24.644/0.808	24.919/0.756	24.993/0.745	…	25.026/0.735	11.327/0.7040	25.086/0.732	25.108/0.731
	$W_5/L_5=W_8/L_8$ [μm/μm]	15.581/1.466	15.458/1.492	15.671/1.444	15.713/1.431	…	15.742/1.429	18.379/0.565	15.782/1.413	15.825/1.408
	W_6/L_6 [μm/μm]	29.797/0.295	29.343/0.27	30.093/0.304	30.174/0.316	…	30.312/0.325	33.6060/0.570	30.465/0.331	30.548/0.341
	W_7/L_7 [μm/μm]	23.873/1.524	23.956/1.499	23.821/1.556	23.77/1.565	…	23.76/1.571	33.8530/0.861	23.72/1.576	23.706/1.58
	C_C [pF]	**12.259**	11.936/	12.434	12.504	…	12.538	10.362	12.563	12.635
	C_L [pF]	14.419	14.581	14.314	14.241	…	14.227	16.002	14.171	**14.138**
	I_{bias} [μA]	**27.481**	27.36	27.727	27.818	…	27.855	26.581	27.930	27.981
Design Specifications	DC gain [dB]	71.683	70.311	72.087	72.615	…	72.967	**75.63**	73.221	73.637
	GBW [MHz]	**2.341**	2.3776	2.332	2.337	…	2.331	2.327	2.335	2.328
	Phase margin [deg]	**54.252**	56.21	53.307	52.183	…	51.505	50.129	50.914	50.164
	Slew rate [V/μs]	2.666	2.5009	2.661	2.661	…	2.659	2.103	**2.668**	2.661
	Output swing [V]	2.327	2.3263	2.326	2.326	…	2.327	**2.371**	2.327	2.327
	CMRR [dB]	81.494	81.229	81.502	81.598	…	81.837	77.220	82.123	**87.416**
	$PSRR^+$[dB]	73.705	72.638	74.085	74.561	…	74.882	**83.413**	75.149	75.570
	$PSRR^-$ [dB]	100.460	97.850	100.200	101.310	…	102.500	85.003	102.450	**103.460**
Objectives	Area [μm^2]	153.174	152.4001	153.964	154.010	…	154.337	**122.13**	154.624	155.260
	Power consumption [mW]	0.789	0.816	0.780	0.769	…	0.763	0.940	0.757	**0.748**
TOI		0.1778	0.1548	0.1606	0.1540	0.1516	…	0.1500	0.1485	**0.1492**

Table 7 Optimal design of parameters, specifications, objectives, and TOI for MOPSO algorithm.

MOPSO		Pareto-solutions						
		1	2	3	4	5	6	7
Design Parameters	$W_1/L_1=W_2/L_2$ [μm/μm]	7.568/1.367	8.411/1.464	14.107/1.332	6.484/1.416	12.924/1.34	8.627/1.391	10.094/1.301
	$W_3/L_3=W_4/L_4$ [μm/μm]	22.511/1.279	10.496/1.508	24.642/0.989	19.366/1.272	19.866/1.111	20.276/1.271	26.725/0.808
	$W_5/L_5=W_8/L_8$ [μm/μm]	15.726/1.281	8.375/1.119	13.405/1.227	13.649/1.26	9.786/1.27	12.678/1.334	12.955/1.203
	W_6/L_6 [μm/μm]	37.749/0.563	22.054/0.449	21.915/0.305	33.489/0.548	18.986/0.31	29.183/0.375	27.811/0.285
	W_7/L_7 [μm/μm]	18.418/0.964	22.921/0.67	20.317/1.08	16.599/0.919	15.438/0.797	10.584/0.642	19.126/1.191
	C_C [pF]	11.135	8.964	11.634	10.845	9.428	12.278	12.31
	C_L [pF]	9.486	10.042	9.531	9.181	8.55	8.016	9.151
	I_{bias} [μA]	22.162	37.668	21.724	23.889	36.864	25.352	22.845
Design Specifications	DC gain [dB]	**79.296**	72.729	74.067	78.197	72.445	72.867	71.756
	GBW [MHz]	2.113	3.443	2.463	2.091	**4.049**	2.172	2.190
	Phase margin [deg]	51.274	58.045	51.755	54.470	52.557	55.667	**59.140**
	Slew rate [V/μs]	2.213	**4.568**	2.083	2.436	4.237	2.266	2.074
	Output swing [V]	**2.365**	2.308	2.352	2.355	2.304	2.365	2.343
	CMRR [dB]	**108.677**	93.156	98.652	99.358	91.661	78.619	81.534
	PSRR$^+$[dB]	**86.172**	79.378	76.093	85.245	75.113	80.875	73.791
	PSRR$^-$[dB]	**121.63**	87.097	118.88	115.7	93.453	101.03	102.73
Objectives	Area [μm^2]	157.573	**100.286**	147.845	135.632	121.825	127.105	131.327
	Power consumption [mW]	**0.566**	1.698	0.617	0.641	1.180	0.664	0.662
TOI		0.0980	0.2037	0.1063	**0.0975**	0.1776	0.1046	0.1082

Table 8 Optimal design of parameters, specifications, objectives, and TOI for NSGA-II algorithm.

NSGA-II		Pareto-solutions								
		1	2	3	4	…	17	18	19	20
Design Parameters	$W_1/L_1=W_2/L_2$ [μm/μm]	10.143/1.046	12.808/1.853	10.425/1.034	12.491/1.633	…	7.185/1.336	7.185/1.336	7.188/1.336	7.184/1.336
	$W_3/L_3=W_4/L_4$ [μm/μm]	16.988/0.908	32.126/0.925	24.704/0.908	32.123/0.925	…	19.761/0.521	19.756/0.523	19.767/0.522	19.765/0.524
	$W_5/L_5=W_8/L_8$ [μm/μm]	16.164/0.701	16.403/1.104	16.196/0.779	16.403/1.046	…	12.408/1.036	12.408/1.048	12.408/1.021	12.408/1.055
	W_6/L_6 [μm/μm]	29.756/0.324	29.751/0.325	29.755/0.325	29.751/0.325	…	24.578/0.377	24.583/0.377	24.585/0.377	24.584/0.377
	W_7/L_7 [μm/μm]	33.364/0.437	33.371/0.436	33.364/0.437	33.370/0.436	…	15.694/0.895	15.694/0.89	15.694/0.795	15.694/0.872
	C_C [pF]	11.246	10.948	10.96734	10.948	…	9.755	9.771	9.741	9.748
	C_L [pF]	13.053	10.135	11.04778	10.135	…	9.108	9.095	9.117	9.094
	I_{bias} [μA]	21.4758	20.8225	20.99641	20.840	…	21.7796	21.77965	21.77966	21.77966
Design Specifications	DC gain [dB]	70.382	**71.024**	70.322	70.966	…	70.42	70.388	70.365	70.301
	GBW [MHz]	2.511	2.205	2.567	2.295	…	2.504	2.544	**2.651**	2.525
	Phase margin [deg]	57.922	**58.471**	55.802	57.237	…	57.712	57.467	52.651	55.102
	Slew rate [V/μs]	2.318	2.160	2.284	2.176	…	**2.295**	2.326	2.163	2.309
	Output swing [V]	2.412	2.413	2.413	**2.413**	…	2.412	2.412	2.413	2.412
	CMRR [dB]	77.201	**108.906**	79.472	107.212	…	77.622	77.869	105.039	80.101
	PSRR$^+$[dB]	76.172	**76.403**	75.405	76.241	…	76.099	75.898	75.133	76.147
	PSRR$^-$[dB]	**88.421**	81.464	83.833	81.688	…	87.184	86.454	80.991	88.012
Objectives	Area [μm^2]	**99.042**	167.392	115.943	158.843	…	101.401	103.899	146.002	119.513
	Power consumption [mW]	0.825	**0.583**	0.693	0.587	…	0.801	0.779	0.592	0.678
TOI		0.1075	0.1212	0.1050	0.1163	…	0.1036	0.1039	0.1096	**0.1032**

Table 9 Statistical comparison of objective values and TOI of the methods.

		NSGA-II	MOPSO	MOIPO	MOGWO	MOLA
Best	Area	99.042	100.286	122.1317	85.8733	**72.825**
	Power Consumption	0.583	0.566	0.748	0.573	**0.560**
	TOI	0.1032	0.0975	0.1451	0.0710	**0.0526**
Worst	Area	167.392	157.572	155.261	234.091	**81.563**
	Power Consumption	0.825	1.180	0.940	0.674	**0.650**
	TOI	0.1212	0.2037	0.1606	0.1778	0.0567
Mean	Area	125.875	131.656	147.807	112.583	**75.499**
	Power Consumption	0.686	0.861	0.811	**0.614**	0.625
	TOI	0.1081	0.1280	0.1526	0.0879	**0.0558**
Variance	Area	536.942	341.846	159.312	1783.500	**6.663**
	Power Consumption	0.008	0.179	0.004	9.6333E-04	**8.5793 E-04**
	TOI	2.9815E-05	0.0019	2.0063E-05	9.4183E-04	**1.7106 E-06**

Fig. 9 shows the Pareto-fronts in the best run (in terms of TOI criterion) for the proposed methods. Despite the greater spread of the Pareto-front of the NSGA-II, it can be argued that MOLA responses have dominated Pareto-front solutions of other algorithms. The HSPICE simulation results obtained from the optimally designed two-stage CMOS op-amp are shown in Figs. 10-13 for the best solution (based on the best TOI value) in the best run. Also, the results obtained from the MOLA method are shown in the figures. A comprehensive comparison is presented in Table 10 between the results of the proposed MOLA algorithm and those of other rival methods along with other studies. Finally, for the performance analysis of Pareto indexes and runtime of MOLA with other assumed algorithms for the best run, Table 11 is provided.

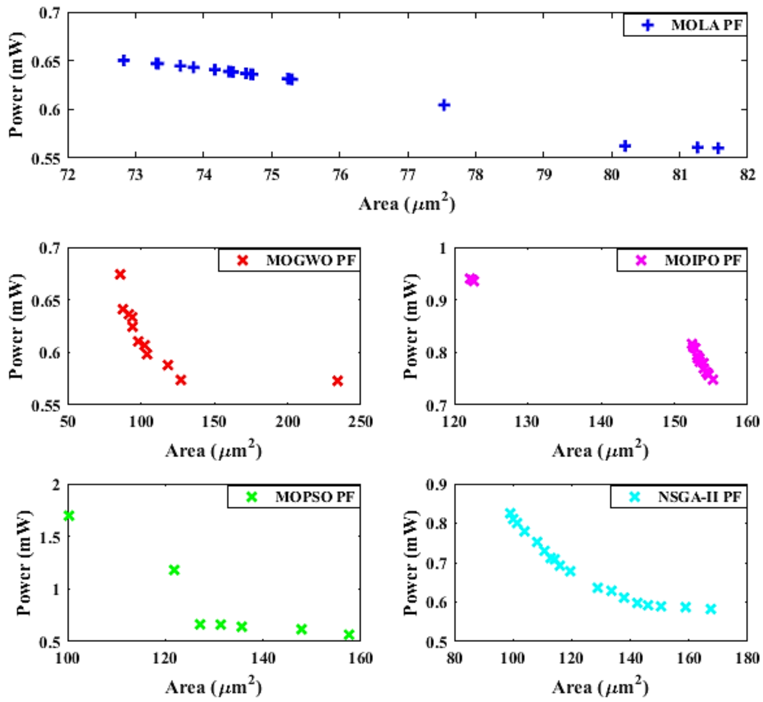

Fig. 9 Pareto-front of the proposed methods.

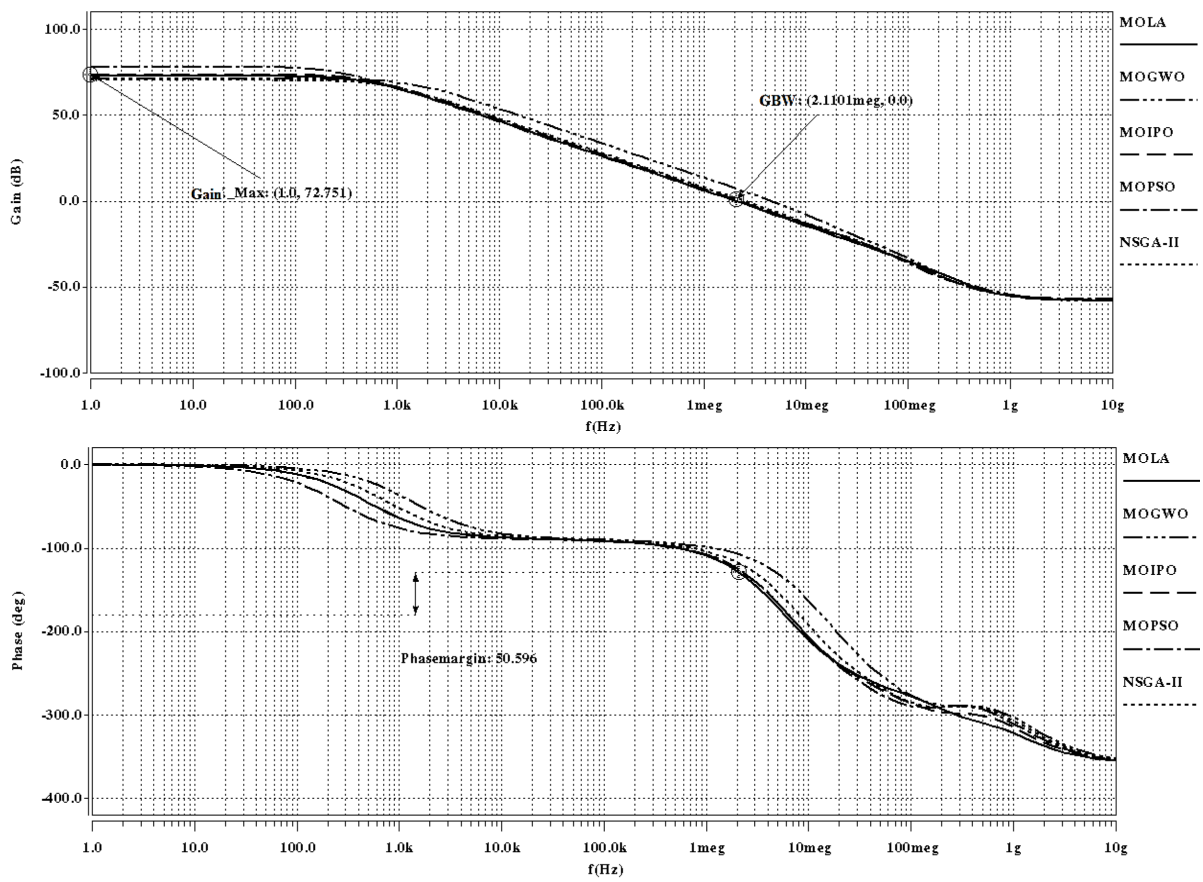

Fig. 10 Bode diagram plotted by the proposed methods.

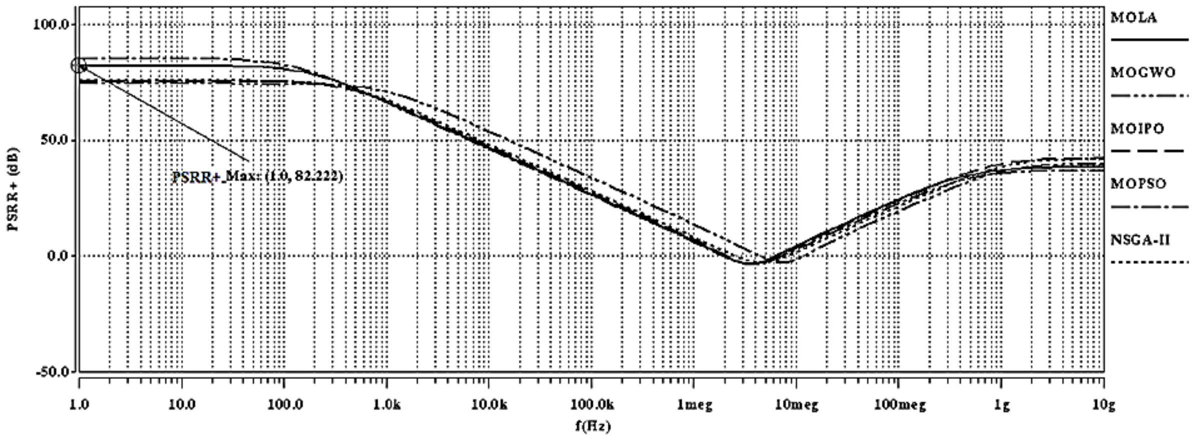

Fig. 11 Positive PSRR of plotted by the proposed methods.

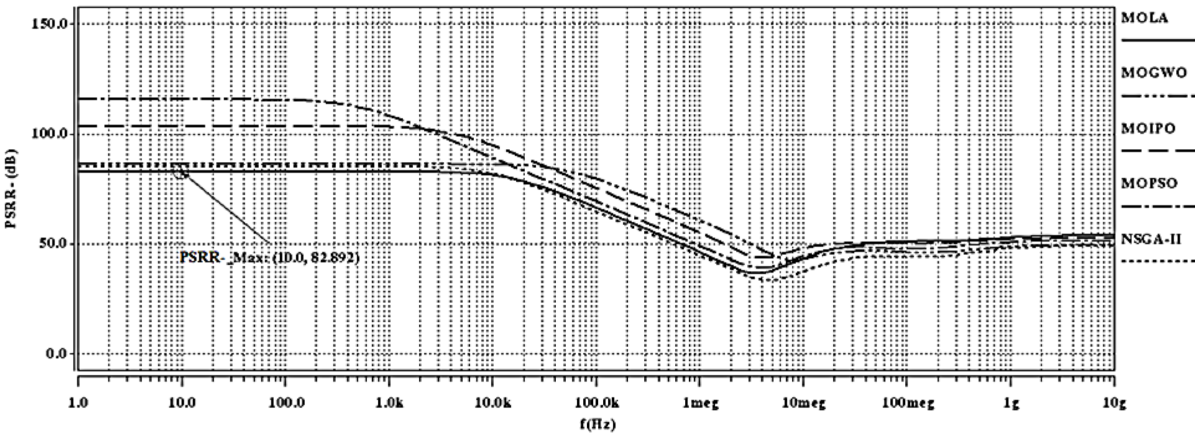

Fig. 12 Negative PSRR of plotted by the proposed methods.

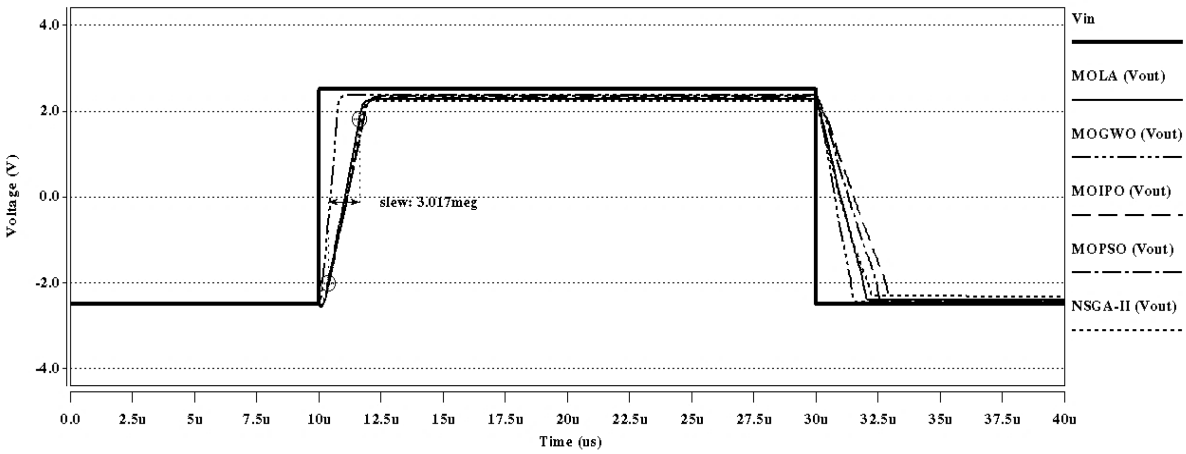

Fig. 13 Slew rate of plotted by the proposed methods.

Table 10 Compare the best results with previous works.

Parameters	References			Present work				
	GSA-PSO [7]	AGSA_PSO+PF [8]	CO-GSA [9]	NSGA-II	MOPSO	MOIPO	MOGWO	MOLA
Technology [μm]	0.35	0.25	0.25	0.25	0.25	0.25	0.25	0.25
DC gain [dB]	75.43	70.441	74.785	71.024	**79.296**	75.63	74.121	73.808
GBW [MHz]	**5.776**	2.017	2.644	2.651	4.049	2.340	4.722	2.127
Phase margin [deg]	66.2	50.181	**78.448**	58.471	59.140	54.252	53.962	60.616
Slew rate [V/μs]	10.88	2.231	**10.897**	2.295	4.568	2.668	5.725	3.023
Output swing [V]	-	**2.415**	2.232	2.413	2.364	2.371	2.355	2.343
CMRR [dB]	87	88.187	78.040	**108.906**	108.677	87.416	80.341	104.156
$PSRR^+$ [dB]	83.2	72.675	**87.190**	76.403	86.172	83.413	77.867	82.530
$PSRR^-$ [dB]	110.4	**131.910**	86.650	88.421	121.630	103.460	92.281	95.485
Area [μm^2]	109.6	210.003	129.845	99.042	100.286	122.13	85.873	**72.825**
Power consumption [mW]	0.713	0.701	0.349	0.583	0.566	0.748	0.573	**0.560**
TOI	0.1330	0.3908	0.2400	0.1032	0.0975	0.1451	0.0710	**0.0526**

Table 11 Pareto and timing performance analysis.

Parameters	NSGA-II	MOPSO	MOIPO	MOGWO	MOLA
SP	21.695	27.155	**0.197**	6.666	0.9305
ONVG	**20**	7	16	11	**20**
Time [s]	1248	1436	1456	1356.2	**1188.86**

5 Conclusions

In this paper, for the first time, the workability of learning automata verified in the optimal design of analog circuits. The circuit was a two-stage CMOS op-amp as a challenging and complex engineering problem. The optimized circuit provided the following features: simultaneous optimization of area and power consumption, minimizing the TOI, satisfies of design characteristics. The performance of the proposed MOLA method with four rival optimization algorithms NSGA-II, MOPSO, MOIPO, and MOGWO on the designed circuit has been investigated comprehensively. Results obtained by MOLA shown the significant improvement of the desired features in terms of the best Pareto-fronts along with suitable evaluation criteria. As future work, we will apply the proposed methodology to optimize more complex analog and digital circuits with particular design specifications. Also, optimization algorithms and reinforcement learning methods can be combined to make the circuit more efficient.

References

[1] S. H. Mirhosseini and A. Ayatollahi, "A low-voltage, low-power, two-stage amplifier for switched-capacitor applications in 90 nm CMOS process," *Iranian Journal of Electrical and Electronic Engineering*, Vol. 6, No. 4, pp. 199–204, 2010.

[2] Ž. Rojec, Á. Bűrmen, and I. Fajfar, "Analog circuit topology synthesis by means of evolutionary computation," *Engineering Applications of Artificial Intelligence*, Vol. 80, pp. 48–65, 2019.

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

[4] C. K. Maiti, and T. K. Maiti, *Strain-Engineered MOSFETs*.CRC Press, 2012.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197, 2002.

[6] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, Vol. 3, No. 2, pp. 95–99, 1988.

[7] S. Mallick, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization," *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 1, pp. 309–331, 2017.

[8] M. Dehbashian and M. Maymandi-Nejad, "A new hybrid algorithm for analog ICs optimization based on the shrinking circles technique," *Integration, the VLSI Journal*, Vol. 56, pp. 148-166, 2017.

[9] M. Dehbashian and 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*, Vol. 59, pp. 218-232, 2017.

[10] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 65, No. 6, pp. 1954-1967, 2018.

[11] S. Dash, D. Joshi, A. Sharma, and 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*, Vol. 94, No. 1, pp. 27–47, 2018.

[12] S. Dash, D. Joshi, and G. Trivedi, "Multi-objective analog/RF circuit sizing using an improved brain storm optimization algorithm," *Memetic Computing*, Vol. 10, No. 4, pp. 1–18, 2018.

[13] E. Afacan, "Inversion coefficient optimization based analog/RF circuit design automation," *Microelectronics Journal*, Vol. 83, pp. 86–93, 2019.

[14] M. Hasanzadeh-Mofrad and A. Rezvanian, "Learning automata clustering," *Journal of Computational Science*, Vol. 24, pp. 379–388, 2018.

[15] M. Ahangaran, N. Taghizadeh, and H. Beigy, "Associative cellular learning automata and its applications," *Applied Soft Computing*, Vol .53, pp. 1–18, 2017.

[16] B. Damerchilu, M. S. Norouzzadeh, M. R. Meybodi, "Motion estimation using learning automata," *Machine Vision and Applications*, Vol. 27, No. 7, pp. 1047–1061, 2016.

[17] N. Kumar, J. H. Lee, and 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*, Vol. 16, No. 3, pp. 1148–1161, 2015.

[18] A. L. Bazzan, "Aligning individual and collective welfare in complex socio-technical systems by combining metaheuristics and reinforcement learning," *Engineering Applications of Artificial Intelligence*, Vol. 79, pp. 23–33, 2019.

[19] M. Rezapoor Mirsaleh and M. R. Meybodi, "Balancing exploration and exploitation in memetic algorithms: a learning automata approach," *Computational Intelligence*, Vol. 34, No. 1, pp. 282–309, 2018.

[20] W. Li, E. Özcan, and R. John, "A learning automata based multiobjective hyper-heuristic," *IEEE Transactions on Evolutionary Computation*, Vol. 23, No. 1, pp. 59–73, 2017.

[21] F. Hourfar, H. J. Bidgoly, B. Moshiri, K. Salahshoor, and A. Elkamel, "A reinforcement learning approach for waterflooding optimization in petroleum reservoirs," *Engineering Applications of Artificial Intelligence*, Vol. 77, pp. 98–116, 2019.

[22] M. L. V. Tsetlin, *Automaton theory and modeling of biological systems*. New York: Academic Press, pp. 160–196, 1973.

[23] H. L. Liao and Q. H. Wu, "Multi-objective optimization by learning automata," Journal of global optimization, Vol. 55, No. 2, pp. 459–487, 2013.

[24] H. Faraji Baghtash and Kh. Monfaredi, "A novel active feedback frequency compensation scheme for two-stage OTA," *Iranian Journal of Electrical and Electronic Engineering*, Vol. 15, No. 3, pp. 321–329, 2019.

[25] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 256–279, 2004.

[26] A. Mohammadi, M. Mohammadi, and S. H. Zahiri, "Design of optimal CMOS ring oscillator using an intelligent optimization tool," *Soft Computing*, Vol. 22, No. 24, pp. 8151–8166, 2017.

[27] S. A. Mirjalil, S. Saremi, S. M. Mirjaliliand L. D. S. Coelho, "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, Vol. 47, pp. 106–119, 2016.

[28] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, pp. 1942–1948, 1995.

[29] M. H. Mozaffari, H. Abdy, and S. H. Zahiri, "IPO: an inclined planes system optimization algorithm," *Computing and Informatics*, Vol. 35, No. 1, pp. 222-240, 2016.

[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

N. Sayyadi Shahraki received the B.Sc. degree in Electrical Engineering from Shahrekord University, in 2010 and the M.Sc. degree in Electrical Engineering from the 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, electronic circuit design.

S. H. Zahiri received the B.Sc., M.Sc. and Ph.D. degrees in Electronics Engineering from Sharif University of Technology, Tehran, Tarbiat Modarres University, Tehran, and Mashhad Ferdowsi University, Mashhad, Iran, in 1993, 1995, and 2005, respectively. Currently, he is a 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.