ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Reinforcement Learning-Driven Co-Optimization Technique for RF Circuit Parameterss

Yuchen Mu[1,*] *and Zhen Tian*[2]

[1] School of Materials Science and Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, 030025, China
[2] James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

SUMMARY: *The optimization process of RF circuits usually needs to address the complexity of high-dimensional nonlinear problems. To cope with this challenge, this paper applies reinforcement learning techniques to it and proposes a DQNN-based collaborative optimization algorithm for RF circuit parameters. The algorithm establishes a mathematical model for the parameter optimization problem of RF circuits, searches for the optimal circuit parameter design in the circuit parameter design space using reinforcement learning algorithms, and designs a DQNN neural network structure to realize the parameter optimization search. Take the charge pump phase-locked loop as experimental example, the method that this paper puts forward shows very good performance on both convergence speed and relative loss. After the optimization work was completed, the phase noise of the voltage-controlled oscillator and the charge pump phase-locked loop has been decreased by 2.14dBc/Hz@1MHz and 4.05dBC/Hz@1MHz, respectively. The tuning range of the voltage-controlled oscillator and the working bandwidth of the charge pump phase-locked loop have respectively seen an increase of 0.122 GHz and 0.034 GHz. Furthermore, the peak vibration frequency of the voltage-controlled oscillator has been increased by 0.095GHz@4V. The results show that the method possesses efficient and stable parameter optimization capability, which verifies its applicability in RF circuit parameter co-optimization.*

KEYWORDS: *reinforcement learning; DQNN; parameter optimization; RF circuits*

1 Introduction

Following the rapid progress of modern communication technology and radio-frequency (RF) systems, the requirements for the working performance of RF circuits are continuously increasing. In complex communication and radar applications, RF circuits not only have to meet the basic frequency response requirements, but also need to take into account multiple performance indicators, such as gain, noise figure, bandwidth, power consumption, etc [1]. Along with technology having continuous advancement, the growth of radio-frequency (RF) circuits is now facing more and more difficult challenges.In these complex application scenarios, how to achieve high efficiency and high accuracy while satisfying multiple mutually constraining performance objectives has become one of the core issues in RF circuit design [2]. At present, the vast majority of RF circuits are designed manually by design experts, requiring designers to build appropriate circuit topologies according to performance requirements, use simplified circuit models to calculate manually, using knowledge or

*18844399319@163.com
https://doi.org/10.65102/is2026589
Published: 30 April, 2026

experience to get device parameters, and using simulators to carry out repeated circuit simulation and correction, this method has no high efficiency. This method cannot catch up with the fast development of the microelectronics industry. Furthermore, it has the situation that automatic design and optimization methods are lacked for RF analog integrated circuits. The non-existence of these automatic design and optimization methods, together with the shortage of auxiliary design tools, has become a bottleneck that hence will hinder the future development of integrated circuits[3-7]. Although most EDA (Electronic Design Automation) tools have integrated basic optimization functions, which can be used to assist in solving design problems or fine-tuning parameters [8]. However, when faced with high-dimensional design variables or multi-objective optimization problems, the optimization capabilities of traditional EDA tools are often limited, especially in nonlinear circuits and complex system architectures, and often fail to compute satisfactory output results [9, 10]. Therefore, for obtaining higher design accuracy and reducing the work burden of designers, it is necessary that we introduce more advanced optimization methods to solve the problems of EDA tools in aspects of accuracy, efficiency, dependence on manpower and function.

For solving this problem, automation design methods which depend on optimization algorithms have appeared and step by step developed into an important research direction in the field of RF circuit parameter design. Single-objective optimization is the thing that uses a special circuit target as the only goal of the optimization problem, and many circuit parameter optimization problems using single-objective algorithms have been studied [11].As an example, Zhou, R and other persons [12]have brought forward a rule-guided genetic algorithm (RG-GA) with the purpose of the optimization of simulated circuit parameters. When we compare with the random variation method that is used in traditional genetic algorithms, RG-GA has a design rule-directed variation (RGM) mechanism. This mechanism makes that the solution region can be searched in a more straightforward way.Instead of giving the optimization work of circuit parameters to a completely mathematical algorithm, this method uses precious design experience to raise the search efficiency. Wu, S et al [13] proposed a method for automatic generation and optimization of LNA circuit topologies, which achieves efficient circuit topology generation and parameter optimization by means of a library of predefined building blocks (PBBs) and a rule-based non-dominated sequential genetic algorithm (RG-NSGA-II). Afacan, E [14] proposed an inverse coefficient (IC) based optimization method for sizing parameters of analog/RF circuits, this thing offers complete and exact information with regard to the whole saturation area. It also together optimizes the transistors to consider the deviation of the circuits, hence therefore promoting the design efficiency. Fan, Z et al [15] proposed linear adaptive hyperparameter based modified PSO applied to power amplifier, SLMBA (Sequential Load Modulation Balanced Amplifier) designed using the proposed method has saturated drain efficiency (DE) between 62.4% and 71.6%. Fan, Z and Cai, J [16] proposed a particle swarm optimization algorithm with nonlinear adaptive hyperparameter control (PSO-NSH) for single-objective optimization of ultra-wideband power amplifiers, which adjusts the inertia weights and acceleration coefficients through the introduction of a Sigmoid function, and the optimization convergence is smoother, and the design objective meets the output power of more than 40.5 dBm in the range of 0.5-3.6 GHz The design goal is to meet the output power over 40.5 dBm in the 0.5-3.6 GHz range, and the efficiency over 60.4%.

RF circuits generally need to compromise between various parameters when designing, and when using a single-objective optimization algorithm it is generally necessary to select one of the indicators as the optimization objective, and the other indicators are added as constraints [17]. While using multi-objective optimization algorithm, multiple indicators can be directly set as the optimization objective to find the compromise solution.As one example,

Joshi, D and other researchers [18] have completed the development of one combined multi-objective optimization framework (MHPSO). They have completed this work through the combination of the particle swarm optimization method and the simulated annealing algorithm. Our aim was to carry out the optimization of the circuit parameter design working procedure. After that, the effect of this method was proved in the exploration of performance space of three electronic circuit products. Elmeligy, K and Omran, H [19] proposed a weight-adjustment based multi-objective optimization method for wideband noise-cancellation LNAs, which balances the priorities of different objectives by adjusting the weight assignment of each optimization objective (e.g., gain, bandwidth, noise figure of merit, and etc.), and allows to find the optimal design point that satisfies the different constraints on the Pareto front in the design space. Touloupas, K and Sotiriadis, P [20] have brought forward a Local Constraint Multi-Objective Bayesian Optimization method (LoCoMOBO) for the automatic size calculation and exploration of performance balance in analog and RF integrated circuits. This method can effectively carry out the work of searching for a Pareto-optimal solution inside a trust region which belongs to the search domain. This result is obtained by it through the way of local Gaussian process modeling and a multi-objective framework of Bayesian optimization. Therefore, it has the big effect of decreasing the time which is needed for parameter optimization.

Reinforcement learning, which is one method of machine learning, puts its core on the interaction that is between an intellectual agent and its environment. It endeavors to obtain the most excellent decision-making methods by a procedure of repeated attempt and mistake study, with the aim of making the cumulative discounted repayments reach maximum value [21].Its theory basic is the Markov Decision Process, which is namely MDP, that is constituted by the state space, the action space, the state transition function, the reward function, and the discount factor [22]. In order to improve design efficiency and optimize circuit parameters, RF circuit design based on reinforcement learning techniques has become a cutting-edge direction to break through the limitations of traditional design methods by transforming key decision optimization problems in RF circuit design into reinforcement learning problems [23]. Take as an example, Hosny, A and other persons [24] have brought forward a reinforcement learning optimization framework that is for circuits with AIG format. Inside this frame, the parameter optimization of one circuit is by us modeled as a Markov Decision Process (MDP). The state space of this MDP includes the main features of the AIG. These features include the main input items, the main output items, the amount of nodes, the quantity of edges, the number of logic layers, the quantity of latches, and the ratio of AND gates to non-AND gates.Pasandi, G et al [25] Q-Learning based reinforcement learning approach for exact logic synthesis which utilizes reinforcement learning for optimization at the process mapping stage, proposes a Hamming distance based metric for evaluating the error rate of approximation logic circuits, Reinforcement Learning agent optimizes the area and latency by adapting the circuit structure while ensuring that the error rate is maintained within the acceptable range. Choi, Y et al [26] proposed an analog circuit optimizer (MA-Opt) based on reinforcement learning, which optimizes multiple predictions of the circuit design through multiple actuators to achieve accelerated optimization of the circuit parameters, and introduced a synergistic near-sampling method, which exploits synergistic effects to achieve optimal design. Zhu, K et al [27] proposed an optimization framework combining GCN (Graph Convolutional Networks) and Reinforcement Learning, which is also aimed at optimizing the parameters of logic circuits in AIG format. This study explored the structural features of the circuits through GCN and used Reinforcement Learning to make decisions for logic optimization. Mirhoseini, A et al [28] combined deep reinforcement learning to the global layout design optimization problem of chip circuits and proposed an end-to-end

learning method for macro cell placement, which models the layout problem of chip circuits as a sequential decision making problem to find a more optimal circuit layout solution by designing an intelligent body capable of acquiring deep features of the chip netlist.

For the purpose of promoting the effect of parameter design concerning radio-frequency (RF) circuit systems, this thesis puts forward a collaborative optimization algorithm for RF circuit parameters which is established on reinforcement learning. On the basis of RF circuit parameter optimization problem modeling, the use of reinforcement learning algorithms in the RF circuit parameter design space to search for the optimal circuit parameter design, the use of DQNN neural network for rapid simulation verification of the search results, and the results will be fed back to the reinforcement learning so as to guide it to search for the RF circuit parameter design that meets the target requirements. One example research is conducted by making use of a charge-pump phase-locked loop. The purpose is to carry out optimization on the voltage-controlled oscillator inside the charge-pump phase-locked loop through the utilization of the method this paper puts forward. After that, a comparison is conducted by us among the phase noise, maximum vibration frequency, and adjustable scope of the voltage-controlled oscillator that is before and that is after the optimization procedure. Furthermore, an analysis upon the changes of the phase noise and output bandwidth of the charge-pump phase-locked loop has been carried out. The purpose of these steps lies in confirming whether the optimization method we put forward can realize the simultaneous optimization of many target parameters.

2 RF circuit parameter co-optimization algorithm

With the development of electronic components toward smaller and denser, the parameter design space of high-speed electronic circuits becomes more and more huge, and how to efficiently and quickly design electronic circuits to meet the target signal integrity requirements becomes more and more important. Based on reinforcement learning techniques, this paper proposes a collaborative optimization algorithm for RF circuit parameters.

2.1 Enhanced learning

Reinforcement learning, which is an often used algorithm, has uses in many different situations like control, decision-making, and recommendation. This tool possesses the capability of dealing with problems which can be generalized as Markov decision-making processes. This algorithm is constituted by three key important components which are state s, action a and reward r. The strategy π_t refers to the action that the intelligent body can take based on the state at this point in time, and the iterative training phase is carried out in which the strategy π is constantly being adjusted in real time based on feedback. Once the policy is determined, the intelligent body can know the probability distribution of executing each action in this state, as shown in equation (1):

$$\pi_t(a \mid s) = p\left(a_t \mid s_t\right) \tag{1}$$

From the foundation, inside an environment that includes both rewards and punishments, the goal of reinforcement learning is to find the most appropriate action for each state of the system. This target is reached by lasting training and adjustment, hence the goal is to make the accumulated average reward get maximum value. This algorithm is constituted by two main composition parts: the intelligent agent and the environment. In state s_t, the intelligent body

will perform the action a_t, and the environment will generate a reward r_t based on the action performed to be fed back to the intelligent body, while transferring to the next state s_{t+1}. By continuous exploration and mutual action with the environment, the intelligent body step by step changes the strategy that it picks in every single situation. At the end, it obtains the most appropriate strategy which satisfies the given standards.

For finding the strategy which can give the biggest total reward, therefore, we have the necessity to give definition to the cumulative reward function. Because the problem we now face satisfies the Markov property, the definition of the accumulated reward R_t can be shown in equation (2):

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2}$$

where $\gamma \in [0,1]$ is used as a discount factor to weigh future rewards against current rewards. If γ is set relatively small, it means that the intelligence will pay more attention to the current reward. If the value of γ is set closer to 1, this point shows that the reason will put more emphasis on the future repayment. In order to find the optimal strategy, a suitable discount factor γ needs to be set.

For the carrying out of accurate assessment on the environment's internal state and the effect of implemented actions.Two value functions are defined based on two different inputs. The state value function $V^{\pi}(s)$ if the input is the state s and the action value function $Q^{\pi}(s,a)$ if the input is the state-action pair $\langle s,a \rangle$. The state-valued function $V^{\pi}(s)$ is expressed as the reward expectation of the intelligent body for executing the current strategy π while in state s. The definition of this function is shown in equation (3) below:

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{\pi}\left[R_t \mid S_t = s\right] \\ &= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s\right] \\ &= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V^{\pi}\left(s'\right) \mid S_t = s\right] \end{aligned} \tag{3}$$

where $\mathbb{E}(\cdot)$ is denoted as the reward expectation under the strategy π and s' is denoted as the next state.

Similarly, the action-value function $Q^{\pi}(s,a)$ is denoted as the goodness or badness of performing the action a at state s. The definition of this function can be expressed in equation (4):

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbb{E}_{\pi}\left[R_t \mid S_t = s, A_t = a\right] \\ &= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s, A_t = a\right] \\ &= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V^{\pi}\left(s'\right) \mid S_t = s, A_t = a\right] \end{aligned} \tag{4}$$

Based on the observation of Eqs. (3) and (4), it is possible to both represent the state value function $V^{\pi}(s)$ and the action value function $Q^{\pi}(s,a)$ using recursion. After describing

the problem to be solved as an MDP process, it is also possible to use the Bellman equation to represent these two value functions. By utilizing such value function expressions as in Eqs. (3) and (4), the maximum expectation for each state and state-action can be obtained. As long as the optimal $V^{\pi}(s)$ and $Q^{\pi}(s,a)$ have been obtained, all the unknown variables included in the MDP process can be obtained. From the observation of using the above method to obtain the optimal policy π^{*} for reinforcement learning, it is possible to define the optimal state-value function $V^{*}(s)$ and the optimal action-value function $Q^{*}(s,a)$ in terms of Eqs. (5) and (6), respectively:

$$\pi^{*} \to V^{*}(s) = \max_{\pi} V^{\pi}(s) \tag{5}$$

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{6}$$

Based on the definition which belongs to the optimal value function, we can clearly get that when this function keeps unchanged, it shows that the optimal strategy for every single state has already been gotten. Through the utilization of the Bellman equation to calculate the optimal value function with respect to the problem which we currently handle, the optimal policy can be obtained through determination π^{*} is then found.

2.2 Modeling the Parameter Optimization Problem

The RF circuit parameter co-optimization problem refers to the automatic search in the design space of the parameters to meet the requirements of the signal integrity indicators of the parameter design. Assuming that $x \in \mathbb{R}^{d}$ represents d RF circuit parameters to be designed, and for each RF circuit parameter x_i, its value range is $\left[x_i^{\min}, x_i^{\max}\right]$, for a set of RF circuit parameters x corresponding to the performance metrics of signal integrity analysis is denoted by $y = f(x)$. The signal integrity optimization objective is divided into three cases: greater than or equal to the target metric $(y \geq \tau)$, less than the target metric $(y < \tau)$, and within the target metric interval $\left(y \in [\tau_1, \tau_2]\right)$, then the signal integrity of RF circuitry gain is expressed as follows:

$$\mathrm{FOM}(y) = \begin{cases} y - \tau & y \geq \tau \\ \tau - y & y < \tau \\ \dfrac{1}{2}(\tau_2 - \tau_1) - \left| y - \dfrac{1}{2}(\tau_1 + \tau_2) \right| & y \in [\tau_1, \tau_2] \end{cases} \tag{7}$$

In the present research paper, we utilize the maximization problem to be an illustration (the minimization problem can be converted into a maximization problem through the negation of the objective function). The optimization question of RF circuit parameters may be put as a boundary-restricted optimization problem:

$$\begin{aligned} &\text{maximize} \quad \mathrm{FOM}(y) \\ &\text{s.t. } c_i(y) \geq 0 \\ &\forall i \in 1 \ldots N_c \end{aligned} \tag{8}$$

where $c_i(y) \geq 0$ denotes the ith constraint.

2.3 DDQN parameter optimization algorithm

2.3.1 MDP Parameter Optimization Problem Description

The Markov Decision Process (MDP) may be characterized through the tuple $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$, where at each decision point t, the intelligent first observes the state $s_t \in \mathcal{S}$ and selects the action $\pi(\mathcal{S} \to \mathcal{A})$ to select the action $a_t \in \mathcal{A}$ and then transfer the probability according to the state:

$$p\left(s_{t+1} \mid s_t, a_t\right) \in P(\mathcal{S} \times \mathcal{A} \to \mathcal{S}) \tag{9}$$

Into a new state s_{t+1}. At the same time, an immediate reward value $r_t \in R(\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R})$ serves as a feedback for the state transition. Here, the RF circuit parameter optimization problem is mapped into an MDP:

ENVIRONMENT: In the work items of radio-frequency (RF) circuit design, the usual working environment includes circuit netlists and industrial-level circuit simulation software. In this research paper, one pre-trained deep neural network model is utilized as the replacement for the industrial circuit simulation software. Furthermore, a data processing component has been added to quickly process data, which includes such operations as normalization and inverse normalization.

State: In the RF circuit parametric design environment, the circuit itself and the specification of each parameter is the main area of observation. Thus, the state of the intelligent body at the moment t is denoted as:

$$s_t = \left[x_1, x_2, \ldots, x_d\right] \tag{10}$$

where x_i denotes the ith RF circuit parameter.

Action: a discrete action space is used to adjust the parameters of the circuit. For each tunable circuit parameter x_i, there are three possible actions at moment t: increase $\left(x_i + \Delta\right)$, hold $\left(x_i + 0\right)$, and decrease $\left(x_i - \Delta\right)$, where Δ is the smallest unit for updating the parameter and the range of values is limited to $\left[x_i^{\min}, x_i^{\max}\right]$.

Reward:The building of the reward numerical value is intricately connected with the optimization target of the RF circuit. For engineering a more high-performing radio frequency circuit, the present reward function t is defined as:

$$r_t = \delta \cdot \mathrm{FOM}_t \tag{11}$$

where δ is a constant factor used to adjust the size of the reward value. To allow for more situations to be explored, r_t is also set to a larger constant at certain moments. The final goal

that intelligence pursues is to find out the most beneficial strategy π^* by maximizing the cumulative MDP reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] \tag{12}$$

2.3.2 Algorithm flow

Just like what the previous section has proved, in the model of Markov Decision Process (MDP), actions are defined as discrete variables, hence the dimension of the action space has a limit. Therefore, this paper uses the dual deep Q-network (DDQN) reinforcement learning algorithm for the optimization of circuit parameters. Figure 1 shows the structure of the RF circuit parameter co-optimization algorithm which is based on DDQN.The Double Deep Q-Network (DDQN) algorithm has inside it two deep neural networks that share a same structure: the evaluation network and the objective network. These two deep neural networks each carry out work with no dependence on the other. This independent operation has the function of eliminating the correlation between the estimated value function of one action and the target value function. In addition, the DDQN algorithm realizes the separation of action choice and assessment through the use of two different estimators, which effectively reduces the number of the action parameters. This paper deals with the problem of over-valuation which occurs in deep Q-network (DQN) algorithms, that therefore causes the results to deviate from the optimal value, hence enhancing the effect of reinforcement learning.

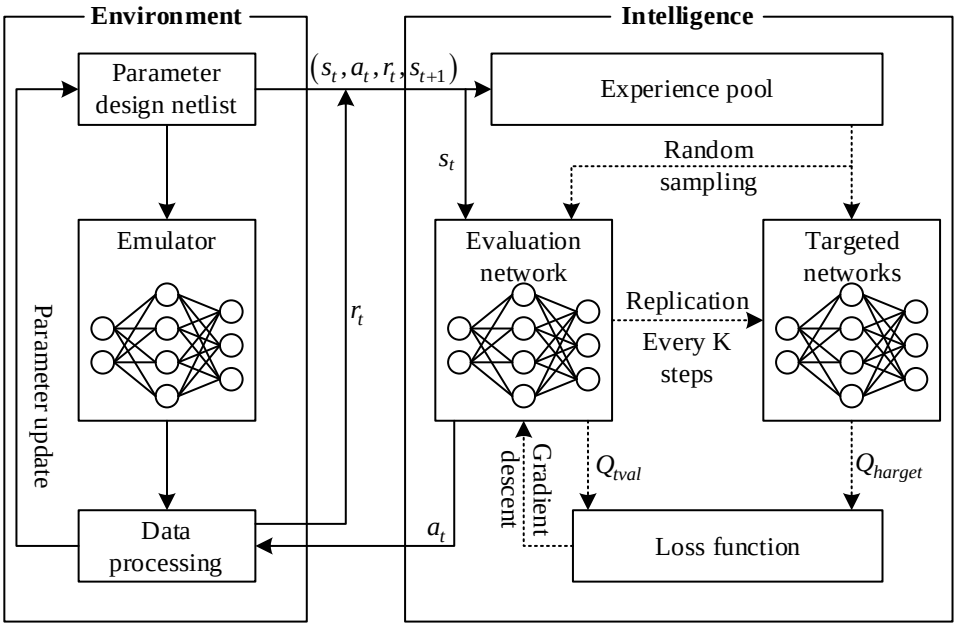

Figure 1: The framework of the Rf circuit parameter optimization algorithm based on DDQN

When we carry out decision-making concerning the optimization question of circuit parameters, the algorithm abides by the following concrete working step:

(1) Initialization: initialize the evaluation network Q and its parameters θ, the objective network Q' and its parameters $\theta^- = \theta$, the experience pool P and its capacity N.

(2)Action judgement: the wisdom body obtains the circuit input parameter x_0 as the initial state s_t, and selects the action a_t according to the $\epsilon - greedy$ strategy:

$$a_t = \begin{cases} \arg\max_a Q(s_t, a; \theta) & \text{There is a } 1-\epsilon \text{ probability} \\ random & \text{Otherwise} \end{cases} \tag{13}$$

The numerical value of ϵ experiences attenuation along with the proceeding of iteration quantities. This characteristic makes the model be able to carry out random exploration of a broader scope of strategies in the beginning phases. Along with the increase of iteration number, the model is more inclined to select the most optimum tactic. π^* based on the previous experience (evaluating the output of the neural network).

(3) Environment interaction: when you carry out the current action, the environment will deal with it to get a new condition s_{t+1}, the already trained simulation neural network model $f_{FOM}(\cdot)$ can quickly get the current RF circuit parameter design x_t corresponding to the signal integrity metrics y, and the reward value r_t can be obtained according to Eq. (7) and Eq. (8). Store the quaternion $\langle s_t, a_t, s_{t+1}, r_t \rangle$ into the experience pool P, P can be realized by using the data structure queue, and when P reaches the maximal capacity N, the old data stored in P is deleted.

(4) Calculate the target value: m samples $\langle s_j, a_j, s_{j+1}, r_j \rangle$ are randomly sampled from P, furthermore, the target numerical value is calculated by means of the following formula:

$$Y_j = \begin{cases} r_j & \text{Reaching the termination state} \\ r_j + \gamma Q'\left(s_{j+1}, \arg\max_a Q(s_{j+1}, a; \theta); \theta^- \right) & \text{Otherwise} \end{cases} \tag{14}$$

where $\gamma \in (0,1)$ is the discount factor used to reduce future rewards.

(5) Parameter update: The parameters θ of the evaluation network are updated by minimizing the loss function (gradient descent):

$$L(\theta) = \frac{1}{m} \sum_{j=1}^{m} \left(Y_j - Q(s_j, a_j; \theta) \right)^2 \tag{15}$$

At intervals of K steps, the parameters of the target network θ^- are updated by copying the parameters of the evaluation network θ:

$$\theta^- = \theta \tag{16}$$

2.3.3 DDQN Neural Network Structure

DQN utilizes a greedy algorithm to obtain Q values, Q values are prone to overestimation during the computation process, which leads to a decrease in the accuracy of the algorithm.Double-Deep Q-Network (DDQN) is constituted by two different networks: the evaluate network and the goal network. These two network structures are utilized by us for the measurement of the values. The goal network acts as an assisting network, which helps the assessment network to reach convergence and reduce overestimation.

In Double Deep Q-Network (DDQN), the evaluation network acts as a function which computes the value of Q every possible action, just as is shown by the following equation.

$$Q_\pi\left(s_t, a_t; w(t)\right) = E\left[\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau} \mid s_t, a_t\right] \tag{17}$$

where $w(t)$ represents the network parameters, $r_{t+\tau}$ is the reward from a_t, and $\gamma \in [0,1]$ is the discount factor controlling the effect of future rewards $r_{t+\tau}, \tau \geq 1$. Based on the function values $Q_\pi\left(s_t, a_t; w(t)\right)$, the action a_t proceeds with probability as shown below:

$$\pi\left(a_t \mid s_t\right) = \begin{cases} 1-\varepsilon_t + \dfrac{\varepsilon_t}{|A|}, & a_t = \arg\max\limits_{a \in A} Q_\pi\left(s_t, a; w(t)\right) \\ \dfrac{\varepsilon_t}{|A|}, & a_t \neq \arg\max\limits_{a \in A} Q_\pi\left(s_t, a; w(t)\right) \end{cases} \tag{18}$$

Drive the environment to move randomly to the next state s_{t+1}. After storing $\left(s_t, a_t, r_t, s_{t+1}\right)$ into the experience pool, enter the time step $t+1$ and repeat the above steps. Note that in order to finally obtain a deterministic mapping from s_t to a_t, the value of ε_t should lie in the interval $[0,1)$ and decrease as t increases.

Theoretically, given a quaternion of experience bars stored in the experience pool, at all time steps $t' \leq t$, the evaluation network should update its parameters w so that $Q_\pi\left(s_{t'}, a_{t'}; w\right)$ and $r_{t'} + \gamma \max_{a \in \mathcal{A}} Q_\pi\left(s_{t'+1}, a; w(t')\right)$ are minimized.

However, this may lead to fluctuations in w or even cause overestimation, thus weakening the performance of the algorithm. The emergence of goal networks greatly alleviates this problem. Essentially, the target network is the same function as the evaluation network, except that its network parameters are $w(t - t \bmod c)$ and are updated every c time steps (since $t - t \bmod c$ remains constant every c time steps). The reference network, which is regarded as an archived version of the assessed network, it calculates a time-difference (TD) objective y_t for each time step t:

$$y_t = r_t + \gamma Q_\pi\left(s_{t+1}, a^\star; w(t - t \bmod c)\right) \tag{19}$$

$$a^\star = \arg\max_{a \in A} \gamma Q_\pi\left(s_{t+1}, a; w(t)\right) \tag{20}$$

The evaluation network builds a loss function through making use of the TD target data:

$$L(w) = \frac{\sum_{k=1}^{K}\left(Q_\pi\left(s_{t(k)}, a_{t(k)}; w\right) - y_{t(k)}\right)^2}{2K} \tag{21}$$

where $s_{t(k)}, a_{t(k)}, y_{t(k)}$ are randomly taken out from the experience storage pool to construct a small-scale data batch which has the size k. The evaluation work of this network is conducted via the utilization of the loss function. According to this loss function, the evaluation network's parameters are updated on the basis of equation (22), where α is the learning rate:

$$w(t+1) = w(t) - \alpha \frac{\partial L(w)}{\partial w} \tag{22}$$

When we face the design question of radio-frequency (RF) circuit parameters, there exist very many cases in which people must hold other parameters unchanged to carry out fine adjustments upon one single parameter. Therefore, the output of the value for every selected action is not a requirement. For letting the intelligent systems produce more excellent and more quick best selection decisions, the dueling network framework is utilized in this place. This structural framework divides the output of the network into two different branches, the state value function $V(s)$ and the dominance function $Adv(s,a)$, which are ultimately combined to output the values of the actions:

$$Q(s,a) = V(s) + Adv(s,a) \tag{23}$$

where $V(s)$ measures how good the state is in a given state and $Adv(s,a)$ measures the advantage of picking an action in a given state. The above equation cannot uniquely separate $V(s)$ from $Adv(s,a)$ during neural network training, and the method of setting $Adv(s,a)$ to 0 is used here:

$$Q(s,a) = V(s) + \left[Adv(s,a) - \max_{a'} Adv\left(s,a'\right) \right] \tag{24}$$

Further, the max operation is removed using averaging:

$$Q(s,a) = V(s) + \left[Adv(s,a) - \frac{1}{|Adv|} \sum_{a'} Adv\left(s,a'\right) \right] \tag{25}$$

This can make the network more stable as well as easier to optimize, although it can make $V(s)$ and $Adv(s,a)$ deviate from the objective.

3 Parameter optimization simulation experiment

In the present research article, the parameter optimization of a charge pump phase-locked loop (CPPLL) is selected as a case example to confirm the RF circuit parameter co-optimization algorithm that is based on the Double Deep Q-Network (DDQN).

3.1 CPPLL System Design

3.1.1 CPPLL structure

The charge pump phase-locked loop, which is a circuit that combines digital and analog parts, is constituted by a plurality of digital and analog circuit components. It obtains widespread utilization in the design of frequency sources which are for transmitters.The charge pump phase locked loop system is mainly constituted by five circuits: the phase frequency detector (PFD), the charge pump (CP), the low pass filter (LPF), the voltage controlled oscillator (VCO), and the frequency divider (FD). Figure Two shows the structure of charge pump phase-locked loop.

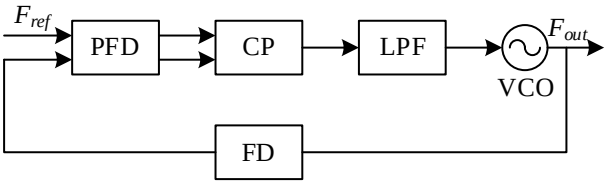

Figure 2: The structure of charge pump phase locking loop

3.1.2 Performance indicators

In a charge pump phase-locked loop, the performance indicators have complex mutual interactions. Therefore, designers must must reach the most appropriate balance between each metric's good points and bad points, at the same time maintaining the overall performance with the greatest effort they can make.

(1) Frequency range (bandwidth) of the output signal

The output signal which is produced by a phase-locked loop is located within a specific frequency range, and the two boundary values of this range are $f_{\min}$ and $f_{\max}$, the magnitude of the output signal's frequency spectrum depends on both the frequency regulation ability of the voltage-controlled oscillator (VCO) and the magnitude of the control voltage.

(2) Lock Time

The locking time length refers to the time that the phase-locked loop requires to change from the capture state to the tracking state. The locking time length of the phase-locked loop can be confirmed through separately observing the change in the control voltage and the change in the output signal.

(3) Phase Noise

Phase noise is regarded as the decibel-based expression of the ratio of the single-sideband noise spectral density to the complete signal power each hertz at a specific offset frequency. This measurement is carried out in the unit of dBc/Hz. The out-band phase noise of a phase-locked loop is in the big part controlled by the phase noise of the voltage-controlled oscillator (VCO).

Through the overall inspection of the key performance indicators of the phase-locked loop, we can clearly see that the working behavior of the voltage-controlled oscillator (VCO) mainly affects the frequency range and phase noise of the output signal of the phase-locked loop. For the promotion of these two properties of the phase-locked loop, it is a necessary thing to carry out careful adjustment on the tuning range and phase noise performance of the VCO.

3.1.3 Design process

Below list the design steps that are for the charge pump phase-locked loop:

(1) Determine the design specifications of the CPPLL

(2) System simulation

Use MATLAB's Simulink to perform system-level simulation of the CPPLL, and determine the design specifications of each module circuit according to the requirements of the CPPLL system specifications.

(3) Design of each module circuit

Based on the design specifications of each module determined by the system in the second step, design and simulate each circuit structure separately, and check whether its performance meets the requirements according to the simulation results. If the result does not meet the requirements, it will continue to optimize it.

(4) Overall circuit simulation

On the basis of completing the module circuit design in the second step, connect each module circuit according to the connection method shown in Figure 2 to form the overall CPPLL circuit and simulate it, and if it does not meet the system specifications, then further optimize the overall CPPLL architecture design in step (2) and the module circuit design in step (3) until it meets the system specifications before it is allowed to flow.

The flow mainly consists of two parts: the generation of simulation data sets and the optimization of DDQN parameter co-optimization algorithm. First, MATLAB is used to establish an equivalent model, randomly generate the initial design parameters, construct 30 groups of design parameters to form a simulation data set, and then optimize it by using the DDQN parameter co-optimization algorithm.

3.2 Algorithm Training Analysis

After 10,000 rounds of training, the convergence curve of Reward during training is shown in Fig. 3, with the dark scattered dots showing the returns obtained at each step and the yellow curve showing the average return for each round of optimization. Beginning from the beginning random arrangement, the strategy step by step gathers together to one that makes the earnings get steady. Because the return function which is defined in the experiment has negative property, a value which is nearer to 0 means a smaller area under the objective function’s convergence curve, hence that means the optimizer has better performance. It is very clear that in the process of 10,000 training times, the Reward value of the convergence curve is getting close to 0.

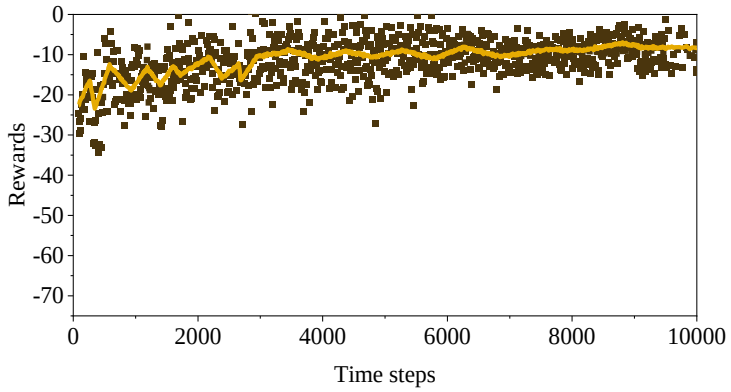

Figure 3: The convergence curve of the reward in training process

Figure 4 gives depiction of the convergence curve that belongs to the parameter optimization algorithm in the scope of the test set. We here give out a comparison, in which the optimization process that uses the differential evolution (DE) algorithm is contained. This figure also gives an exhibition of the comparison between the optimization curve of the DQNN which is put forward in this paper and the optimization curve of the differential evolution algorithm. The Differential Evolutionary Algorithm requires hundreds or even thousands of iterations to find the optimal solution, in contrast, the reinforcement learning approach can find the optimal solution in as short a time as possible due to the learning of a priori information. The final loss function values of the DQNN parameter optimization algorithm based on reinforcement learning and the differential evolutionary algorithm under 30 sets of test data are given in Table 1.The mean values of the number of iterative steps for the DQNN algorithm and the DE algorithm are 6 and 353, respectively, and the mean values of the final loss are 0.0045 and -0.0108, respectively, in addition, both the number of repeated steps and the relative error inside the DQNN parameter optimization algorithm are lower.

Table 1: Iterative steps and final losses of each algorithm

Method	Steps			Relative loss		
	Min	Mean	Max	Min	Mean	Max
DE	2	6	10	0.0032	0.0045	0.0087
DQNN	186	353	600	-0.1200	-0.0108	-0.0495

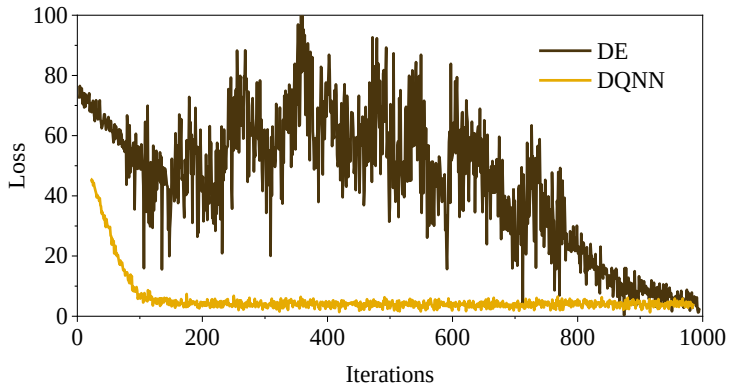

Figure 4: Optimization curves comparison of DDQN algorithms and DE algorithms

Fifteen groups of data are utilized as the training set, thus another fifteen groups of data are utilized as the test set. Support Vector Regression (SVR), K-Nearest Neighbors (KNN), and Random Forest (RF) are employed by us to act as regression models. After that, the prediction results that these regression models got on the test set are compared with the results which are got by the reinforcement learning that is put forward in this paper. We make a comparison on the loss curves of the training set and the test set, which are shown in Figure 5 and Figure 6, which shows that the DQNN parameter joint optimization algorithm is still able to ensure good optimization results. In this figure, the horizontal axis stands for training samples, while the vertical axis shows the magnitude of the loss value of the result which is got by the optimization algorithm and the regression model for this objective function. The nearer the value gets to the real value, the more excellent the result is. When we do consideration on the training set and the test set, the average error between the values got by the DQNN parameter co-optimization algorithm this paper puts forward and the true values is 13.86% and 11.31% separately. With regard to the SVR algorithm, these numerical values are 27.53% and 26.37%. The KNN algorithm displays average error values of 44.97 percent and 46.80 percent, and the RF algorithm possesses error values of 70.63 percent and 72.50 percent. It is very obvious that the DQNN parameter co-optimization algorithm which this paper puts forward can obtain results that are more near to true values.

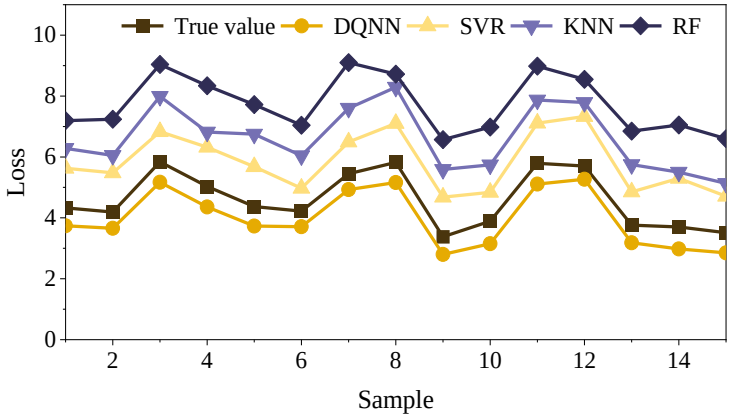

Figure 5: Comparison of loss curves in training sets

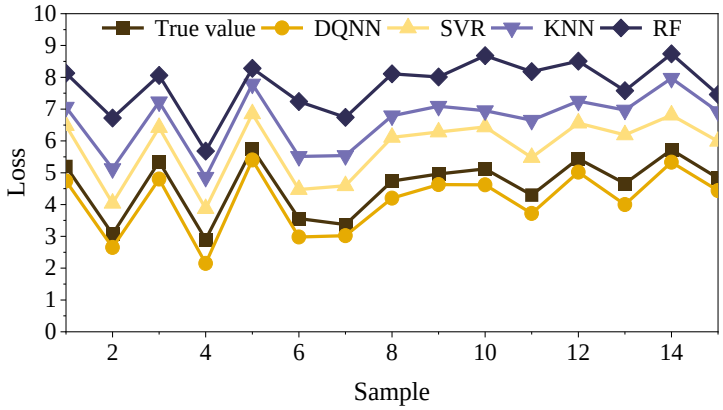

Figure 6: Comparison of loss curves in test sets

3.3 CPPLL Optimization

According to what has been said before, a special trade-off that is between the phase noise and the maximum oscillation frequency of the Voltage-Controlled Oscillator (VCO) exists. For selecting suitable numerical values for the two bias voltages of the variable capacitance device, the minimum phase noise and the maximum oscillation frequency of the VCO are therefore utilized as the objective function of the Dual-Q Network (DQNN) parameter co-optimization algorithm. In this algorithm, the two dimension parts of each single individual stand for the offset electric voltages VA and VB.The VCO which is optimized by the method that is introduced in this section is compared with the SVR algorithm, therefore it showed better performance both before optimization and in the training stage. Figure 7 gives the comparison outcomes about the phase-noise property of the VCO, hence Figure 8 displays the comparison outcomes for the maximal vibration frequency of the VCO.The method which is put forward in this paper can let the simultaneous optimization of phase noise and maximum oscillation frequency performance be finished in the voltage-controlled oscillator (VCO). To speak specifically, it brings about a decreasing of phase noise of 2.14 dBC/Hz at 1 MHz, a increasing of the maximum oscillation frequency by 0.095 GHz at 4 V, and a enlarging of the tuning range by 0.122 GHz.

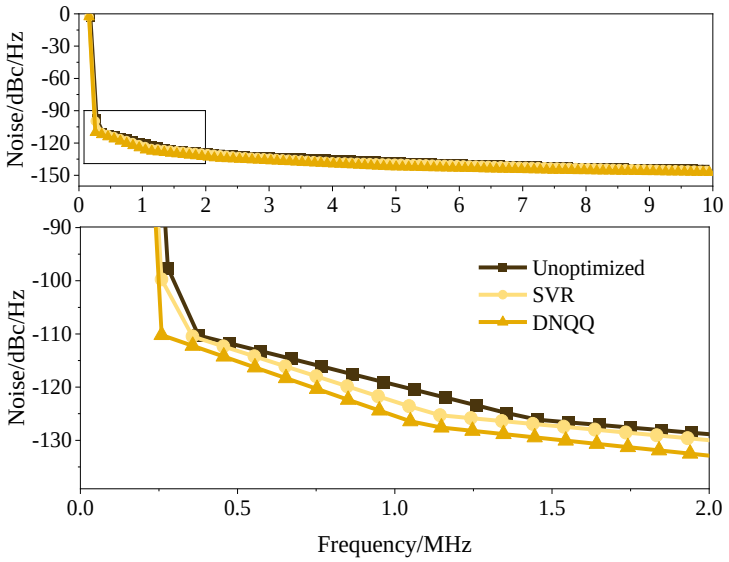

Figure 7: VCO's phase noise performance comparison results

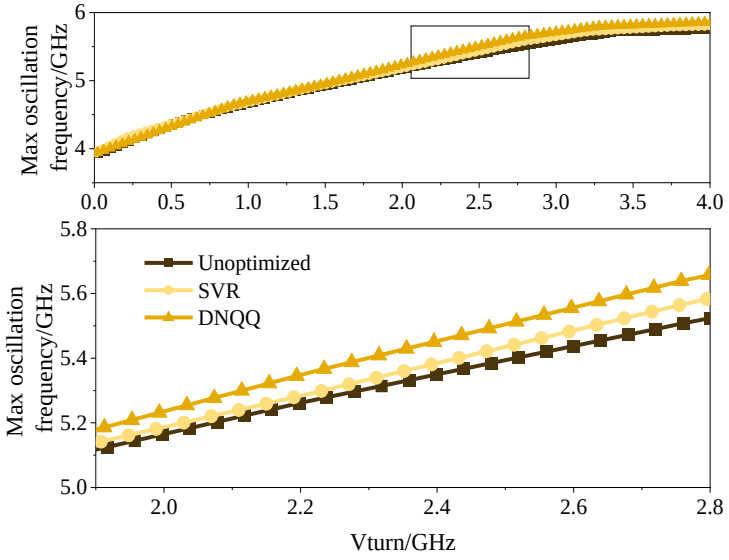

Figure 8: VCO's maximum oscillation frequency performance comparison results

According to the Simulink simulation structure of the CPPLL proposed above, the system simulation of the CPPLL using the optimized VCO is carried out while other structures remain unchanged, furthermore, the entire simulation outcomes of this system are put one beside another with those of the simulation that was done before optimization. The comparison performance result findings of the CPPLL are given in Table 2.The outcome shows that when it is compared with the configuration before optimization, the system phase noise of the CPPLL has been reduced by 4.05 dBC/Hz at 1 MHz, hence the working bandwidth has been increased by 0.034 GHz. After we finished the design of the left modules of the charge-pump phase-locked-loop circuit, the charge-pump phase-locked-loop circuit was carried out test on a flow chip. The final test results, which are showed in Table 2, are in accordance with the design norms that are described in this section.The results of the experiments are basically consistent with the results of the simulations, which thus proves the feasibility of the RF circuit parameter co-optimization method based on DQNN proposed in

the present paper.

Table 2: The performance comparison results of CPPLL

	Pre-optimized CPPLL	Optimized CPPLL	Test results
Bandwidth	0.411GHz	0.445GHz	0.442GHz
Phase noise	-116.28dBC/Hz@1MHz	-120.33dBC/Hz@1MHz	-118.72dBC/Hz@1MHz

4 Conclusion

This paper focuses on introducing reinforcement learning to solve the parameter optimization problem of circuits, and proposes a DQNN-based parameter co-optimization algorithm for RF circuits, and takes charge pump phase-locked ring as an example for experimental simulation and analysis. The main research results are as follows:

(1) The co-optimization arithmetic for RF circuit parameters which this paper puts forward displays a more rapid convergence speed and a lower relative error loss. In both the training set and the test set, the loss number which this method gets is more close to the real value. The deviation between the computation outcome and the actual value is smaller than 14 percent. By comparison, the error values of other comparison algorithms all are above 25%.

(2) Through the optimization which is carried out by the reinforcement learning method that is proposed in this paper, the phase noise of the voltage-controlled oscillator is reduced by 2.14 dBc/Hz when it is at 1 MHz. At the same time, the maximal vibration frequency increases by 0.095 GHz when voltage is 4 V, the adjustable scope enlarges by 0.122 GHz, the noise of charge-pump phase-locked loop decreases by 4.05 dBC/Hz under 1 MHz, and the work bandwidth becomes wider by 0.034 GHz. The present article proves that the method shows outstanding capability in the co-optimization of radio-frequency (RF) circuit parameters.

In this research thesis, we utilize the rules of reinforcement study to change the parameter optimization question into the idea of a surrounding inside reinforcement learning. After we have finished building this environment, we employ a reinforcement learning algorithm to have interaction with it. We then give out rewards to good optimization moving paths and punishments to bad ones. Follow-up research works will focus on enlarging the optimization algorithm, increasing the simulation ability, and simplifying the calculation resources. The purpose is to further promote the application property and practical property of the DQNN algorithm in radio-frequency (RF) circuit optimization tasks.

About the Author

Yuchen Mu was born in Yanji, Jilin. P.R. China, in 2005. I am currently studying at the School of Materials Science and Engineering, Taiyuan University of Science and Technology. My main research direction is Iron and Steel Metallurgy, Metallurgical Physical Chemistry.

Zhen Tian received his bachelor degree in electronic and electrical engineering from University of Strathclyde, Glasgow, UK, in 2020. He is currently pursuing the Ph.D. degree with the James Watt School of Engineering, University of Glasgow, UK. His research interests include safe decision making in autonomous driving, deep reinforcement learning, sensing technologies and control engineering.

References

[1] Yeung, S. H., Chan, W. S., Ng, K. T., & Man, K. F. (2012). Computational optimization algorithms for antennas and RF/microwave circuit designs: An overview. IEEE Transactions on Industrial Informatics, 8(2), 216-227.

[2] Rayas-Sánchez, J. E., Koziel, S., & Bandler, J. W. (2021). Advanced RF and microwave design optimization: A journey and a vision of future trends. IEEE Journal of Microwaves, 1(1), 481-493.

[3] Mina, R., Jabbour, C., & Sakr, G. E. (2022). A review of machine learning techniques in analog integrated circuit design automation. Electronics, 11(3), 435.

[4] Budak, A. F., Gandara, M., Shi, W., Pan, D. Z., Sun, N., & Liu, B. (2021). An efficient analog circuit sizing method based on machine learning assisted global optimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(5), 1209-1221.

[5] Zhang, H. (2025). Physics-Informed Neural Networks for High-Fidelity Electromagnetic Field Approximation in VLSI and RF EDA Applications. Journal of Computing and Electronic Information Management, 18(2), 38-46.

[6] Chang, E., Han, J., Bae, W., Wang, Z., Narevsky, N., Nikolic, B., & Alon, E. (2018, April). BAG2: A process-portable framework for generator-based AMS circuit design. In 2018 IEEE Custom Integrated Circuits Conference (CICC) (pp. 1-8). IEEE.

[7] Jain, N., & Raj, B. (2018). Analysis and performance exploration of high performance (HfO2) SOI FinFETs over the conventional (Si3N4) SOI FinFET towards analog/RF design. Journal of Semiconductors, 39(12), 124002.

[8] Wu, H., Su, Z., Zhang, J., Wei, S., Wang, Z., & Chen, H. (2020). A design flow for click-based asynchronous circuits design with conventional EDA tools. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(11), 2421-2425.

[9] Babu, C. S., Mariyal, M. R., & Mega, A. (2025). EDA Tools and Methodologies: Enabling Efficient IC Design in Digital and Analog VLSI. Exploring the Intricacies of Digital and Analog VLSI, 87-104.

[10] Wang, L. C. (2016). Experience of data analytics in EDA and test—principles, promises, and challenges. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(6), 885-898.

[11] Passos, F., González-Echevarría, R., Roca, E., Castro-López, R., & Fernández, F. V. (2019). A two-step surrogate modeling strategy for single-objective and multi-objective optimization of radiofrequency circuits. Soft Computing, 23(13), 4911-4925.

[12] Zhou, R., Poechmueller, P., & Wang, Y. (2022). An analog circuit design and optimization system with rule-guided genetic algorithm. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(12), 5182-5192.

[13] Wu, S., Li, Y., Tan, T., Huang, Z., Qiao, J., & Li, X. (2025). An Automated Circuit Topology Generation and Optimization Method for CMOS Low-Noise Amplifiers. IEEE Transactions on Circuits and Systems I: Regular Papers.

[14] Afacan, E. (2019). Inversion coefficient optimization based analog/RF circuit design automation. Microelectronics Journal, 83, 86-93.

[15] Fan, Z., Hao, Z., Huang, J., & Cai, J. (2024). Design of rf-input sequential lmba using pso algorithm with improved linear self-adaptive hyper-parameters. IEEE Transactions on Microwave Theory and Techniques, 72(11), 6414-6425.

[16] Fan, Z., & Cai, J. (2024, May). Design and optimization of a multi-octave power amplifier using an improved PSO algorithm with nonlinear adaptive hyper-parameters. In 2024 IEEE MTT-S International Wireless Symposium (IWS) (pp. 1-3). IEEE.

[17] Sağlican, E., & Afacan, E. (2023). MOEA/D vs. NSGA-II: A comprehensive comparison for multi/many objective analog/RF circuit optimization through a generic benchmark. ACM Transactions on Design Automation of Electronic Systems, 29(1), 1-23.

[18] Joshi, D., Dash, S., Reddy, S., Manigilla, R., & Trivedi, G. (2023). Multi-objective hybrid particle swarm optimization and its application to analog and RF circuit optimization. Circuits, Systems, and Signal Processing, 42(8), 4443-4469.

[19] Elmeligy, K., & Omran, H. (2022). Fast design space exploration and multi-objective optimization of wide-band noise-canceling LNAs. Electronics, 11(5), 816.

[20] Touloupas, K., & Sotiriadis, P. P. (2021). LoCoMOBO: A local constrained multiobjective Bayesian optimization for analog circuit sizing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(9), 2780-2793.

[21] Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. Adaptation, learning, and optimization, 12(3), 729.

[22] Ernst, D., & Louette, A. (2024). Introduction to reinforcement learning. Feuerriegel, S., Hartmann, J., Janiesch, C., and Zschech, P, 111-126.

[23] Gao, P., Yu, T., Wang, F., & Yuan, R. Y. (2024). Automated design and optimization of distributed filter circuits using reinforcement learning. Journal of Computational Design and Engineering, 11(5), 60-76.

[24] Hosny, A., Hashemi, S., Shalan, M., & Reda, S. (2020, January). DRiLLS: Deep reinforcement learning for logic synthesis. In 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC) (pp. 581-586). IEEE.

[25] Pasandi, G., Nazarian, S., & Pedram, M. (2019, March). Approximate logic synthesis: A reinforcement learning-based technology mapping approach. In 20th International Symposium on Quality Electronic Design (ISQED) (pp. 26-32). IEEE.

[26] Choi, Y., Park, S., Choi, M., Lee, K., & Kang, S. (2024). MA-opt: Reinforcement learning-based analog circuit optimization using multi-actors. IEEE Transactions on

Circuits and Systems I: Regular Papers, 71(5), 2045-2056.

[27] Zhu, K., Liu, M., Chen, H., Zhao, Z., & Pan, D. Z. (2020, November). Exploring logic optimizations with reinforcement learning and graph convolutional network. In Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD (pp. 145-150).

[28] Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., ... & Dean, J. (2021). A graph placement methodology for fast chip design. Nature, 594(7862), 207-212.