



Analog Circuit Complementary Optimization Based on Evolutionary Algorithms and Artificial Neural Network

Behzad Rajabi¹, Farhad Razaghian^{1*}

¹Electrical Engineering Department, South Tehran Branch, Islamic Azad University, Tehran, Iran.

Abstract

In analog circuit optimization, obtaining optimal point that can satisfy various kinds of specifications is posed as goal of design. Utilization of evolutionary algorithms was introduced as a useful method but speed of convergence and ensure to access optimal point are these methods most challenges. In this paper the Multi-Layer Perceptron (MLP) artificial neural network is applied to access the suitable point appropriate different specifications values of analog circuit. This point used in optimization algorithm to find reliable response. Neural network itself is trained by training database is collected during initial optimization process. The link of HSPICE and MATLAB is used for circuit simulation and evaluation during the process.

Keywords: Analog circuit, Evolutionary algorithm optimization, Cost function, Multi-Layer Perceptron Neural network.

1. INTRODUCTION

Nowadays analog circuit automated design and optimization has received much attention, because analog circuit design deal with various specifications and variables, require consuming long design time and highly skilled expert designer [1-14]. Day by day by development of integrated circuits (ICs), complexity of the circuits design is increased and from commercial aspects, improving design and access to exact optimal points is important. Analog circuit design procedure can be consisting of topology selection and circuit sizing [1]. Most analog circuit sizing problems can be expressed as the multi objective optimization problem (e.g., gain power consumption and etc.) with design variables that are size of components (e.g., transistors dimensions).

Evolutionary optimization algorithms have been represented more appropriate approach in analog circuit optimization. Even though they don't guarantee to find exactly the optimal solution [2]. In analog circuits optimization it should be considered that by changing design specifications values (constraints) , optimization algorithms still be able to access an appropriate response. In previous works one set of values of specifications, considered and optimization had been done to satisfy this values and access to better values for one or two specifications as goals of optimization [1, 3].

The work presented in this paper is a new design approach in analog ICs. This method is able to increase reliability to achieve optimum response. It by using appropriate points (Auxiliary Point) obtained from neural network can guarantee accessing appropriate response even values of design several specifications change. The overall

*Corresponding Author's Email: razaghian@azad.ac.ir

process of this work is depicted in Fig.1. Two data flow are pointed out in this figure. One in the right called Learning Flow; consist of training database and neural network training. To gather training database for neural network training, genetic algorithm (GA) optimization process with stochastic initial point as Primitive Optimization has been used. After collecting database, this data is used to train Neural Network. And another in the left, called Design Flow, trained network is used to estimate points which are proportional different specifications values and then these points are applied to optimization algorithm. Learning Flow is performed once, but it might be possible use several times of Design Flow with various set of values of specifications. In Section 4 the quality of collecting database module and how use Auxiliary Point in optimization process will be discussed.

The structure of the paper is as follow that section 2 reviews the related works and then genetic algorithm (GA) optimization and MLP neural network would be investigated in section 3. Section 4 is the main section of this paper consists of the collecting training database, neural network training and how to use it in optimization. Obtained results for the design of CMOS operational amplifiers and comparison with related works can be seen in Section 5 and finally some conclusions can be found in Section 6.

2. RELATED WORK REVIEW

Circuits' synthesis can be classified into two main categories Knowledge-based and optimization-based [1, 3]. Knowledge-based methods use design equations to determine the design parameters and evaluate its performance [5-7]. Since knowledge-based method uses some approximations and heuristics, this method would reduce the accuracy and also to develop design plan to another circuits and technology need to consume large design time [1, 3]. Second method which is based on using an optimization kernel in the procedure of circuit designing can be divided into three general categories equation-based [8, 9],

simulation-based [10-12] and behavioral- based approaches [13, 14].

In the simulation-based methods, simulation tools are used to evaluate circuit performance then this evaluation is applied in optimization process loop. Accuracy is increased and there is no need to provide equations for different circuits. Because of using simulation tools like HSPICE [15], generality and accuracy are increased and ease of use are reasons to use this method in this work [1]. The literature can be classified in another aspect, some of them have used standard algorithms [4, 16-18], and some of them have tried to modify this algorithms, and also it might be used hybrid algorithms [19, 20]. Efforts have been done to access better performance and reduce time consumption. But also didn't answer that by changing design specifications are they also able to produce optimal solution? In this work suitable points proportional to various values of specifications are used that optimization algorithm would be able to access to better performance and it is able to find optimal solution with changing design specifications values.

3. EVOLUTIONARY ALGORITHM & MLP ARTIFICIAL NEURAL NETWORK

Many engineering problems can be supposed as optimization problems to find optimal statue of the system response. It might be a system with unknown or non-derivative transfer function; therefore classic methods wouldn't be able to solve these problems. But evolutionary optimization algorithms, which are based on biological principles borrowed from nature, can offer a solution. This section first start with definition of search space and fitness function and it will be continued by summarized explanation of Genetic Algorithms (GA) concept, finally a brief description of Multi-Layer Perceptron (MLP) neural networks given. Artificial Neural Networks (ANNs) that are computational model inspired from neurological model of human brain have been used for various applications such as classification, pattern recognition, data analysis, func-

tion approximation and etc [21]. ANNs applications in circuit optimization is used usually as a behavioral approximation system but in this work it will be used to provide suitable point of design variables proportional to various specifications values as Auxiliary Point in GA optimization algorithm [22-24].

3.1 Search Space and Cost Function

Search space could be referred to the all possible solutions that are collected. Sometimes, if suppose N independent input parameters, the dimension of the search space is considered N . With considering Fitness or Costs to the search space as another dimension it will get the $(N + 1)$ dimensional.

Cost function indicates the quality of a solution vector \mathbf{X} . If the system has only one output variable y , cost function will be equal y and if It has more than one output variables, they could be combined into a single value. There are methods that could be used for this combination, one of them is Aggregation Method which in this method the Cost function $C(\mathbf{X})$ as expressed in equation 1 equals the weighted sum of the components that coefficients a_i are weight coefficients and $f_i(\mathbf{X})$ are specifications [18].

$$C(\mathbf{X}) = a_1 f_1(\mathbf{X}) + \dots + a_n f_n(\mathbf{X}) = \sum_{i=1}^n a_i f_i \quad (1)$$

In circuit optimization, some specifications such as Gain when reaches larger values are more favorable and some specifications such as power consumption when reach smaller values are more favorable. With considering this point and since optimizing algorithms are done in order to reduce the cost functions, function $f_i(\mathbf{X})$; ($i=1, 2, \dots, n$) in (1) represents one of the design specifications as expressed in equation 2.

$$s_i^* = s_i / d_i$$

$$\Rightarrow f_i(\mathbf{X}) = \begin{cases} s_i^* & \text{if desired to reach smaller value} \\ 1 / s_i^* & \text{if desired to reach larger value} \end{cases} \quad (2)$$

The parameter S is obtained value and d is desired values (constraints in optimization algorithms) of specification. The advantages of this method are: 1) Simple normalization of specifications values are done, 2) Desired value (constraints) of specification automatically participates in cost function whatever specifications be better than desired values, cost function will be reduced more, that is desirable. 3) This cost function can be used to determine the specifications deviation from the desired values. One of the problems coming to circuits' optimization is measuring the deviation, which is not possible with conventional methods like mean square error (MSE) because in this sort of measuring if exist same deviation in larger value and smaller value these methods report same deviation, which is not proper for circuit optimization whereas in circuits access to one aspect is goal.

3.2. Genetic Algorithm (GA)

This is one of the evolutionary algorithms (EAs) that based on the mechanisms of natural selection which have been developed by John Holland in the 1970s. In summary could be said it consists of four steps: Initial population that often is selected randomly, Evaluation where consists of computing the objective values of the solution, Selection in which the fittest individuals are chosen and Reproduction that creates new individuals from the mating pool by crossover and mutation [4, 18]. It is continued again by evaluation, selection and reproduction until it will be reached one of the stop criteria (e.g., number of generation, time spent and etc.). More details about GA can be found in [25, 26].

3.3. Multi-Layer Perceptron (MLP) Neural Network

MLP neural network consists of set of neurons organized in input, hidden and output layers that layers are interconnected by communication links [21-24]. The links of neurons are associated with weights that dictate the effect on the information passing through them. These weights are adjusted

by the training algorithm and after the network training they will be fixed.

In *feed forward computation*, [21] output of hidden layer nodes are given by Eq.(4). Where w_{ij} are the weight connecting the input i to hidden layer node j , and the input vector $\mathbf{X}=[x_1, x_2, \dots, x_n]^T$ with n dimension is considered. f is the activation function shows the output of a neuron. Equation 5 is sigmoid function, one of the commonly activation functions that might be used. b_j is the bias to the node. Feed forward computation in addition to use in training process, also during the usage of the trained neural model is used. The external inputs are fed to the input neurons (i.e., first layer) and then their outputs are fed to the hidden neurons of the second layer and the outputs of hidden layers are fed to the output layer neurons.

$$h_j = f \left(\sum_{i=1}^n w_{ij} \cdot x_i + b_j \right) \quad (4)$$

$$f(x) = \frac{1}{1 + \exp(x)} \quad (5)$$

The output of the network would be represented as Eq. (6) where w_{jk} are the weights connecting the hidden layer to the output layer and m is the number of hidden layer neurons. The linear function can be used as output layer neurons activation function.

$$y_k = \sum_{j=0}^m w_{jk} \cdot h_j \quad (6)$$

Neural network training algorithms such as Supervised ANN training algorithm uses both input x and desired output data d as sample pairs (\mathbf{X}, \mathbf{D}) to determine the weights. In the training process of the MLP network one of the methods is commonly used, is the back-propagation (BP) algorithm [26] that the weights and biases are adjusted to achieve a minimum mean square error (MSE) between the network output and the target value.

The sample pairs (\mathbf{X}, \mathbf{D}) after collection would be divided into three sets: training data T_r , validation data V , and test data T_e . Training dataset are utilized in training process and are used to update

weights and biases during training. Validation dataset are used to verify the quality of the neural network during training process and also could be used to determine the training process stop criteria. Test dataset are used to independently examine the quality of the trained neural network to determine network performance.

4. ANALOG CIRCUIT COMPLEMENTARY OPTIMIZATION

Main process of circuit optimization that has been done in this work is illustrated in Fig.1 .In Design Flow, first values of specifications (constraints) that are the threshold of desired values, are applied to fitness function (Section 3.1) in optimization process and also these values would be given to the trained Neural Network, and variables appropriate with desired specifications would be generated by Neural Network which are the Auxiliary Points for optimization algorithm. In Fig.1 When algorithm in optimization phase (I) lost variation in population, Auxiliary Point which is an individual of population will be added to optimization process in optimization phase (II). This is the process will produce the final response.

As illustrated in Fig.2, the point that is produce by Neural Network is used once as an initial point which in optimization algorithm, as an individual of initial population. And as was described, once is used as an Auxiliary Point in main optimization process. This provides an opportunity that can be a comparison between this method and the standard algorithm. And show how this method can improve results, and it doesn't need to repeat algorithm, several times to access better result. It refers to this fact, that new method has done this paper, with high probability can access to global optimal point rather than standard algorithm.

The optimization process with initial point would be used as validation of main optimization. These processes are caused to complementary optimization because 1) It provide an evaluation process to verify points are found by this optimization process that at least should access to

point which are found by optimization process with using initial point. 2) With adding one suitable point as an Auxiliary Point to optimization process, it causes to increase convergence speed, moreover could be sure there exists at least one point which satisfies constraints. 3) Preserve the nature of GA algorithm, in the diverse selection of initial point. In this work link of HSPICE and MATLAB [27] was used according to simulation-based method. During the optimization, suggested responses generated by optimization engine (implemented in MATLAB) are given to electrical simulator (HSPICE) that will produce the values of specification.

In learning flow in Fig.1, goal is training Neural Network with suitable training database. The training module is depicted in Fig.3 the MLP Neural Network with collected database, for this work aims in analog circuit optimization, will be trained. And then will be used in main process in design flow, as referred in Fig.1.

As illustrated in Fig.3 the Neural Network inputs are different desired specifications values of analog circuit, and outputs are circuit variables. The training process is done with collected training database. As described in section 3.3 Neural network training algorithm, is Supervised ANN training algorithm which uses both input x (specifications) and desired output d (variables) that are collected with quality which is described in next section 4. The back-propagation (BP) algorithm is used which the weights and biases are adjusted to achieve a minimum error between the network outputs and the target values. Weights and biases are updated each cycle according BP algorithm.

Collecting suitable training database would be important for neural networks performance, that training database should be sufficient and have a proper dispersion in database space. Also to train ANN the training database should be normalized, and it is better that data randomly are fed to ANN in training process.

Because of variety and many data which are used in optimization algorithm process, this process itself would be useful to gather sufficient

and convenient training database. In this work as illustrated in Fig.1 in Learning Flow, GA optimization process with stochastic initial point has been used to collect training database and this data was used in MLP Neural Network learning process. Values of specifications, applied to simulation-based optimization, and as were shown in flow diagram Fig.4 during the optimization, data will be collected until optimization algorithm converges (the domain of population variation is small). In this optimization process, vectors of specifications and design variables will be taken as sample data (X, D) input and desired output.

These data should be modified and prepared to use in neural network training. Therefore first, data which are incompatible (e.g., the negative dB gains) are deleted and then all variables and specifications are mapped to $[-1, 1]$. If total collected sample data aren't sufficient (Amount sample data which is require to proper training) it will be run again. Because of changing initial point at each run it will be possible to collect various data with proper dispersion. Random selections of training database are done and then are fed to MLP neural network training. After training Neural Network, it would be used to provide different variable points appropriate to different desired values of specifications. And by changing values of desired specifications Neural Network still will be able to provide proportional variables.

5. DESIGN EXAMPLE

In this section the new method has been applied on conventional CMOS operational amplifier designing. The purpose is investigating the performance of this method in improvement of standard GA optimization algorithm. The method has been used in this algorithm for optimization a typical Miller-compensated two-stage operational amplifier as a conventional example of analog circuits' optimization problem.

5.1 Miller-compensated Two-stage Operational Amplifier Design

The Miller-compensated two-stage operational

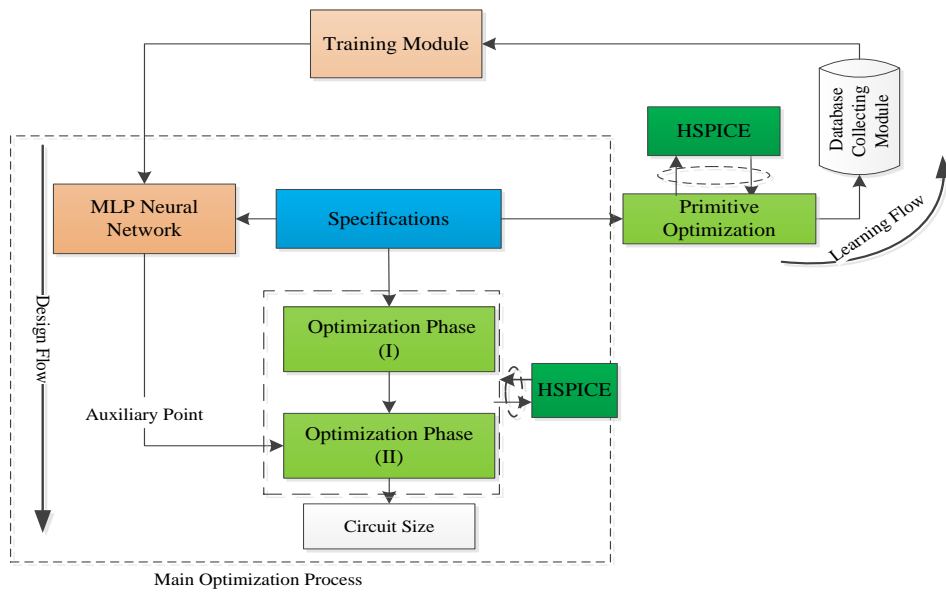


Fig. 1. The overall process.

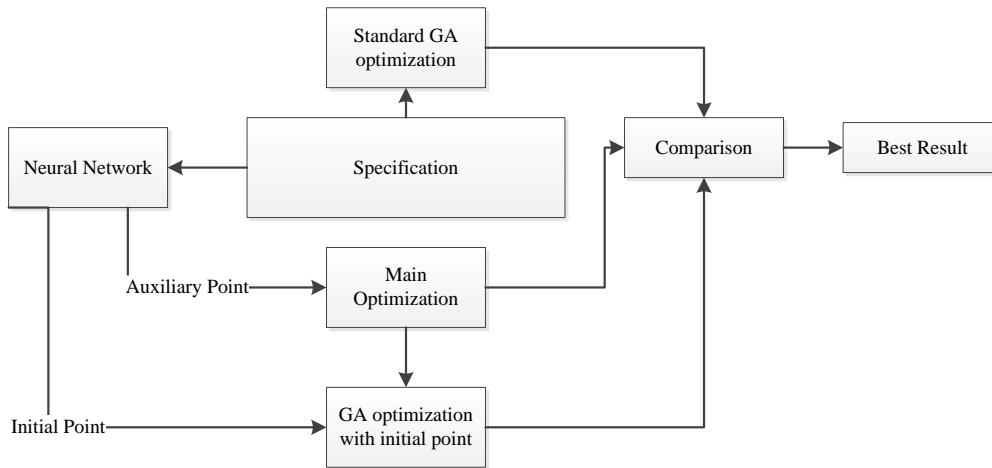


Fig. 2. Use point produce by trained Neural Network.

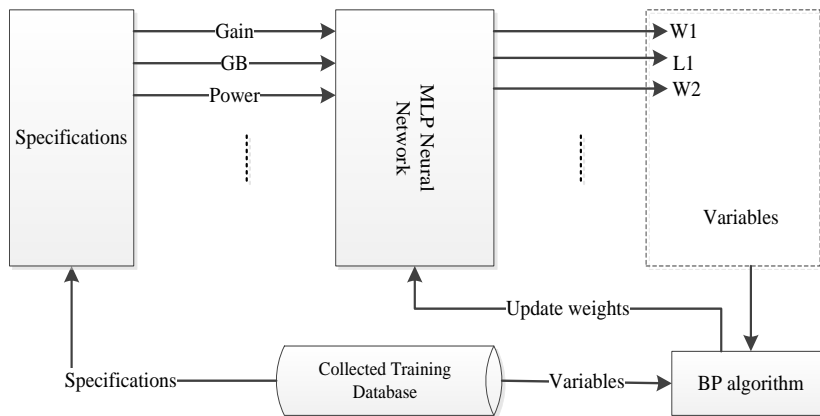


Fig. 3. Training module, neural network training.

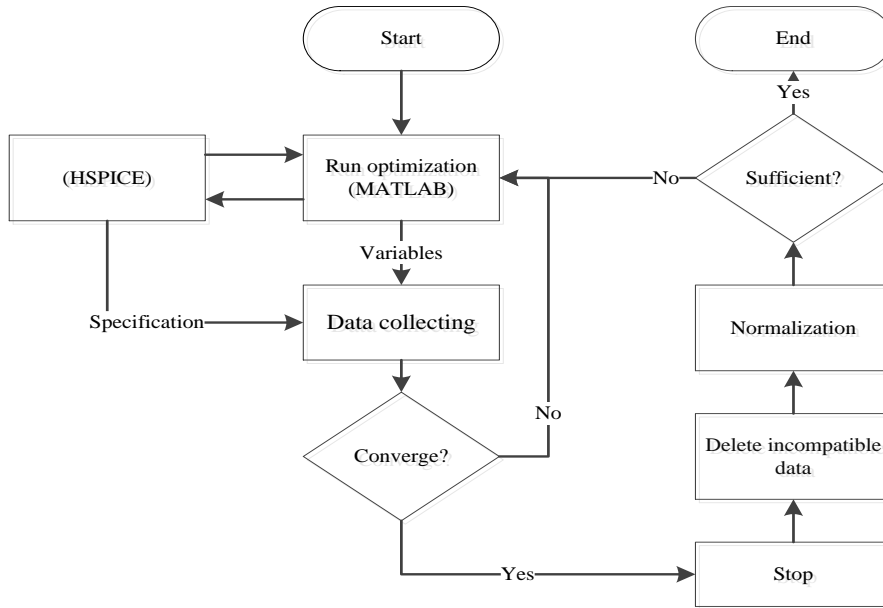


Fig. 4. Flow diagram of collecting training database.

amplifier illustrated in Fig.5 is chosen to test. The technology used is $0.25\ \mu\text{m}$ CMOS process and $C_L=30\text{pF}$. According to the process described in Section 4 to collect training database, first, genetic algorithm with parameters set forth in the table 2 and with fitness function defined in Section 4 has been run 10 times. In each execution, after convergence, algorithm has been stopped and rerun again. In this process after removing inappropriate data 12000 sample data randomly selected.

In this example design variables are: transistor width and length (W , L), bias currents (I_B) and compensation capacitor (C_C) and Specifications and desired values that cloud be considered as constrained are listed in table 1.

The transistor lengths are allowed to change between the minimum values allowed by the technological process and several tens of micrometers. Transistor widths are allowed to change between the minimum technology values and several hundreds of micrometers. Bias currents and capacitor would be changed unreasonable ranges [1].

In this work MLP neural network as illustrated in Fig.6 in overall process (Fig.1) was used. Of course the network was trained with collecting

training database, 6 inputs and 12 outputs. Neural network parameters listed in table 2.

In the training process the 60% of the sample data are assigned to the training set, 20% to the validation set and 20% to the test set. At the end of training Network, the mean square error which obtained after applied test set is $\text{MSE}=0.0603$.

After the completion of network training, to verify the performance the network, different values of specifications are applied to the network and variables are received from the network are applied to the HSPICE and specifications have been received. In table 4 different obtained results and desired values illustrated which can be compared. It shows neural network is able to produce suitable design variables proportional with different values of specification. Design variables related to each column of obtained values of specification listed in table 5.

Constraints are represented in table 1 are applied to the network and the appropriate variables have been received from network as the Auxiliary Point (table 5 column 5) were used in GA algorithms as described in Section 4. The algorithm with parameters listed in tables 2 are used to optimize the Miller-compensated two-stage operational amplifier. After execution the method which was expressed as a main process in section

4, it reached results according table 1 with design variables are listed in table 5

Now, as was expressed in section4 and Fig.2, comparison between this method and the standard algorithm in speed of convergence and reach optimum point could be done.

As illustrated in Fig.7 the speed of convergence could be compared in this method and standard algorithm which shows the method is done in this work are more fast in convergence when after 150 generation reaches acceptable response whereas standard algorithm might need more generation and need to repeat several times. Again as was expressed in Section4 it refers that new method, with high probability able to access to global optimal point which this work method only in 1 run access point which results have listed in table 1 by name WAP. Therefore one of the benefits of this method is not necessary to repeat algorithm several time. In Fig.7 GA-STD is standard form of GA, GA_WAP is GA with Auxiliary Point (this work suggested method) and GA_WIP is GA with initial point.

For a comparison of this method, the results of [2] are shown in the table 1. Results indicate that this method is able to improve the performance. If the cost function represented in section 3.1 with weight coefficients $a_i = 1$ be considered as an indicator of level of achievement to intended specifications and a measure of efficiency. It

would be possible to compare the main method presented in this work with other and standard form of GA optimization algorithm. As listed in Table 1 it can be seen the results in this work is %32 better than standard GA and results in [2] to 29% have improved.

This example total run time in Design Flow is 120.749 second which was run on a Core (i7) 1.73 GHz PC with 4 GB RAM, in the MATLAB environment. Reported include processing time in MATLAB, the simulation time of HSPICE and the communication time between HSPICE and MATLAB.

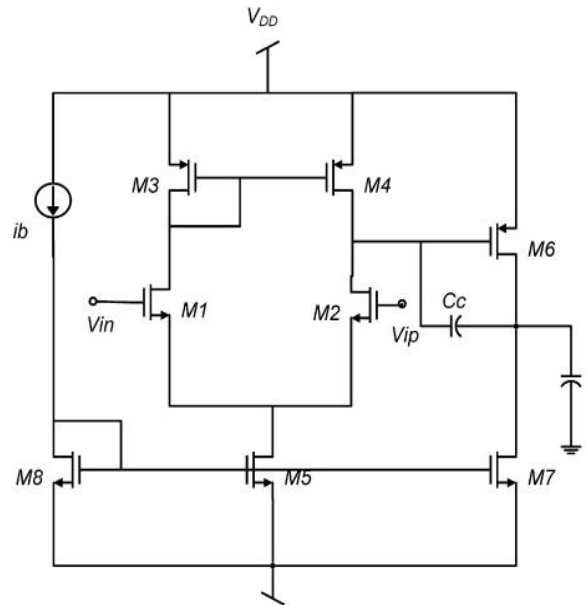


Fig. 5. Miller-compensated two-stage amplifier.

Table 1. Specification obtained result.

| Specifications | Constraints | Target | GA-STD | GA-WAP | Result [1] |
|-----------------------------|-------------|---------|--------|--------|------------|
| DC gain (dB) | ≥ 70 | Maximum | 75.84 | 102.04 | 76.48 |
| GBW (MHz) | ≥ 2 | Maximum | 2 | 6.02 | 2.068 |
| Phase margin ($^{\circ}$) | ≥ 50 | | 80 | 85.01 | 55.96 |
| Slow rate (V/ μ s) | ≥ 1.5 | Maximum | 1.6 | 3.39 | 1.52 |
| Output swing (V) | ≥ 2.5 | | 2.56 | 2.77 | 2.20 |
| Power (mW) | ≤ 2.2 | Minimum | 2.2 | 1.7 | 0.73 |
| Cost Fun | | | 5.46 | 3.72 | 5.23 |

GA_STD is standard form of GA and GA_WAP is GA with Auxiliary Point

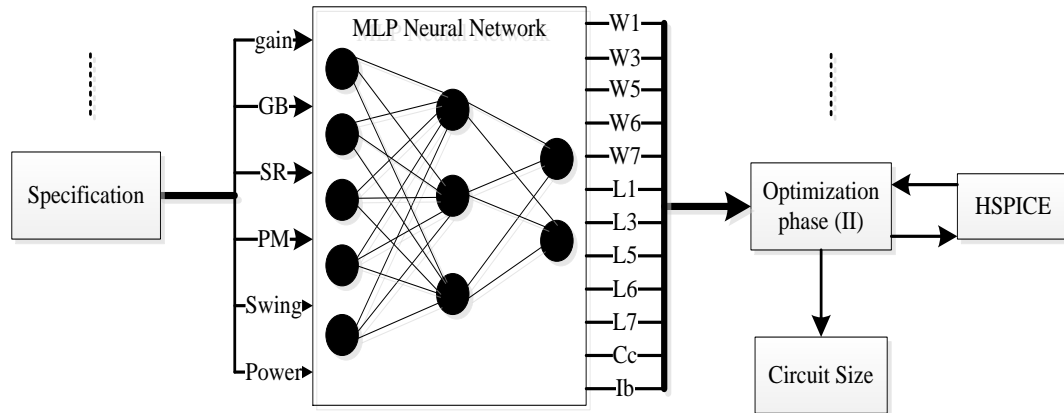


Fig. 6. MLP neural network structure.

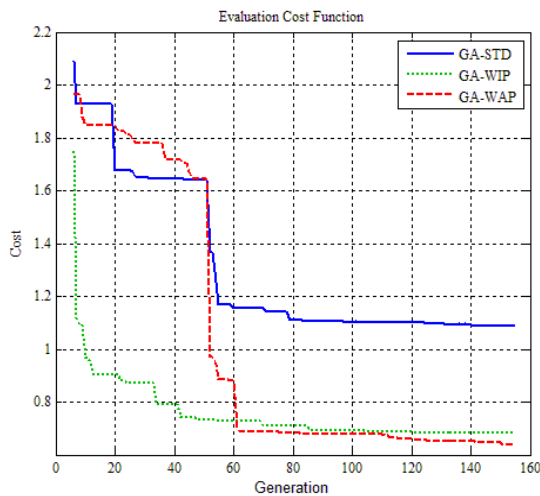


Fig. 7. Evaluation of cost function in GA.

Table 2. Genetic algorithm parameters.

| Algorithm setup | GA-Parameters |
|--------------------|---------------|
| Population Size | 50 |
| Migration Fraction | 0.1 |
| Selection | Random |
| Crossover | scattered |
| Mutation | adaptive |
| Sort | Min. cost |
| Population Type | Double Vector |

Table 3. Parameters of the ANN.

| Parameter | Optimized values |
|-------------------------------|---|
| Architecture | Normal feed-forward MLP |
| Learning rule | Back-propagation (BP) |
| Inputs(N _{in} =6) | Circuit specifications: DC gain (dB), GBW (MHz), Phase margin (°), Slow rate(V/μs), Output swing(V), Power(mW) |
| Hidden layers | 2 layer 18,36 node |
| Outputs(N _{out} =12) | Circuit variables: W1,W3,W5,W6,W7, L1,L3,L5,L6,L7,Cc,ib |
| Network error type | mean square error (MSE) |
| Transfer function | Sigmoid |
| Maximum iterations | 10000 |
| Database size | 12000 |
| Training samples | 7200 |
| Validation samples | 2400 |
| Test samples | 2400 |

Table 4. Evaluating neural network performance in obtaining various values of specifications.

| Specifications | Desired 1 | Obtained 1 | Desired 2 | Obtained 2 | Desired 3 | Obtained 3 |
|------------------------|-----------|------------|-----------|------------|-----------|------------|
| DC gain (dB) | 62 | 62.656 | 95 | 100.352 | 66 | 61.128 |
| GBW (MHz) | 1.2 | 1.174 | 1.5 | 1.312 | 4 | 6.002 |
| Phase margin (°) | 66 | 67.314 | 66 | 68.966 | 82 | 81.052 |
| Slow rate (V/ μ s) | 2 | 1.957 | 3.5 | 3.301 | 2.2 | 2.071 |
| Output swing (V) | 2.6 | 2.688 | 2.5 | 2.455 | 2.5 | 2.452 |
| Power (mW) | 2 | 1.7546 | 3.5 | 2.946 | 4 | 4.304 |

Table 5. Design variable.

| Design variables | Obtained 1 | Obtained 2 | Obtained 3 | Handel point | GA-STD | GA-WAP |
|------------------|------------|------------|------------|--------------|--------|--------|
| W1=W2 (μ m) | 288.935 | 226.567 | 212.764 | 195.802 | 117.61 | 287.35 |
| W3=W4 (μ m) | 258.666 | 230.067 | 209.249 | 117.967 | 276.35 | 223.52 |
| W5 (μ m) | 287.337 | 270.955 | 119.141 | 38.887 | 180.69 | 271.19 |
| W6 (μ m) | 206.745 | 175.776 | 307.564 | 140.538 | 260.00 | 140.54 |
| W7=W8 (μ m) | 190.268 | 135.031 | 258.333 | 200.025 | 37.70 | 266.21 |
| L1=L2 (μ m) | 14.900 | 14.688 | 5.728 | 7.986 | 6.86 | 5.78 |
| L3=L4 (μ m) | 16.044 | 16.553 | 12.955 | 10.632 | 4.67 | 5.79 |
| L5 (μ m) | 14.826 | 15.837 | 21.791 | 11.270 | 8.89 | 4.65 |
| L6 (μ m) | 5.038 | 4.669 | 2.926 | 3.433 | 8.50 | 3.90 |
| L7=L8 (μ m) | 6.674 | 5.638 | 12.088 | 23.398 | 7.04 | 10.00 |
| Cc (pF) | 94.330 | 99.455 | 256.796 | 36.920 | 67.34 | 36.92 |
| Ib(mA) | 0.204 | 0.325 | 0.614 | 0.155 | 0.11 | 0.12 |

6. CONCLUSION

In this paper, a new method in multi-objective optimization technique at circuit sizing problem was presented which uses simulation-based optimization. This method is based on using MLP neural network to provide Auxiliary Point proportional of different desired specifications values which GA optimization method with using these points has been used to operational amplifier parameters optimization.

The results obtained from this method showed that it's able to increase the speed of convergence, with only 150 generation in 150 second. And don't need to run several times which in one run with high probability it's able to reach optimal point. Also by considering cost function as a performance indicator, comparison between this

method and other method and standard form of GA in same condition and same circuit shows %29 and % 32 performance improvements without adding so much complexity to the standard algorithm.

REFERENCES

- [1] B. Liu, Y. Wang, Z. Yu, L. Liu, F. V. Fernandez, "Analog circuit optimization system based on hybrid evolutionary algorithms", Elsevier, the VLSI Journal of Integration, 42(2), 2009, 137-148.
- [2] M. Köppen, G. Schaefer, and A. Abraham, "Intelligent Computational Optimization in Engineering Techniques and Applications", Springer Verlag Berlin, 2011.
- [3] M. Barros, J. Guilherme and N. Horta, "Ana-

- log circuits optimization based on evolutionary computation techniques", Elsevier, the VLSI Journal of Integration, 43(1), 2010, 1365-155.
- [4] R. A. Vural, T. Yildirim, T. Kadioglu, and A. Basargan, "Performance Evaluation of Evolutionary Algorithms for Optimal Filter Design", IEEE Trans on Evol. Computation, 16 (1), 2012, 135 – 147.
- [5] R. Harjani, R. Rutenbar, L.R. Carley, OASYS: a framework for analog circuit synthesis, IEEE Trans. Comput. Aided Des. 8, 1989, 1247–1265.
- [6] N. Jangkrajarn, S. Bhattacharya, R. Hartono, C.-J.R. Shi, IPRAIL" intellectual property reuse based analog IC layout automation", Integration VLSI J. 36 (4) , 2003, 237–262.
- [7] F. El-Turky, E. Perry, BLADES, "an artificial intelligence approach to analog circuit design", IEEE Trans. Comput. Aided Des. 8, 1989, 680–692.
- [8] E. Ochotta, R. Rutenbar, L.R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX, IEEE Trans. Comput. Aided Des. 15, 1996, 273–294.
- [9] M. Del, M. Hershenson, S.P. Boyd, T.H. Lee, "Optimal design of a CMOS Op-Amp via geometric programming", IEEE Trans. Comput. Aided Des. 20 (1), 2001, 1–21.
- [10] A. Torralba, J. Chavez, L.G. Franquelo, FASY,"A fuzzy-logic based tool for analog synthesis", IEEE Trans. Comput. Aided Des. Integrated Circuits 15(7), 1996, 705-715.
- [11] R. Phelps, M. Krasnicki, R. Rutenbar, L.R. Carley, J. Hellums," ANACONDA: simulation-based synthesis of analog circuits via stochastic pattern search", IEEE Trans. CAD 19 (6), 2000, 703–717.
- [12] M. Krasnicki, R. Phelps, R. Rutenbar, L.R. Carley, "MAELSTROM: efficient simulation-based synthesis for custom analog cells", in: Proceedings of the ACM/IEEE Design Automation Conference (DAC), 1999, 945–950.
- [13] G. Alpaydin, S. Balkir, G. Dundar, "An evolutionary approach to automatic synthesis of high performance analog integrated circuits", IEEE Trans. Evol. Comput. 7 (3), 2003, 240–252.
- [14] F. de Bernardinis, M.I. Jordan, A. SangiovanniVincentelli, Support vector machines for analog circuit performance representation, in: Proceedings of the Design Automation Conference, 2003, 964–969.
- [15] Synopsys Inc. HSPICE Simulator. Available: <http://www.synopsys.com>.
- [16] M. S. Bright, T. Arslan, "Synthesis of Low-Power DSP Systems Using a Genetic Algorithm", IEEE Trans. Evol. Comput. 5 (1), 2001, 27–40.
- [17] Y. Jin, M. Olhofer, B. Sendhoff, "A Framework for Evolutionary Optimization with approximate Fitness Functions", IEEE Trans. Evol. Comput. 6(5), 2002, 240–252.
- [18] R.S. Zebulum, M.A. Pacheco and M. Maria, "Evolutionary electronics: automatic design of electronic circuits and systems by genetic algorithms", CRC Press, 2002.
- [19] G. Crina, A. Ajith, I. Hisao, "Hybrid Evolutionary Algorithms. Springer, Heidelberg 2007
- [20] M.A. El-Hosseini, A.E. Hassanien, A. Ajith, A. Al-Qaheri, "Genetic Annealing Optimization: Design and Real World Applications". In: The International Conference on Intelligent Systems Design and Applications, Kaohsiung, Taiwan 2008
- [21] F. Djeflal, M. Chahdi, A. Benhaya, M.L. Hafiane, "An approach based on neural computation to simulate the nanoscale CMOS circuits: Application to the simulation of CMOS inverter", Elsevier, Solid-State Elec. 51, 2007, 48–56.
- [22] Y. Jin, M. Olhofer, B. Sendhoff, "A Framework for Evolutionary Optimization With Approximate Fitness Functions", IEEE Trans. Evol. Comput. 6(5), 2002,

- 240–252.
- [23] G. Wolfe, R. Vemuri “Extraction and Use of Neural Network Models in Automated Synthesis of Operational Amplifiers”, *IEEE Trans. Comput. Aided Des. Integrated Circuits*. 22(2), 2003, 198–212.
 - [24] H. Kabir, Y. Wang, M.Yu, Q.J. Zhang, “High-Dimensional Neural-Network Technique and Applications to Microwave Filter Modeling”, *IEEE Trans. microwave theory and techniques*, 58(1), 2010, 145-156.
 - [25] R. L. Haupt, S. E. Haupt "Practical genetic algorithms" Wiley-Interscience, 2004
 - [26] D.E. Goldberg, K.Deb "A comparative analysis of selection schemes used in genetic algorithm" 1991
 - [27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representation by back-propagation error,” *Nature*, vol. 323, 1986, 533-536.
 - [28] <http://www.mathworks.com/products/matlab>