A FUZZY OPTIMIZATION ENGINE FOR ANALOGUE CIRCUIT DESIGN

G.OLTEAN, S. HINTEA, D. LUPEA
TECHNICAL UNIVERSITY OF CLUJ-NAPOCA, ROMANIA

KEYWORDS: Circuit design, Optimization, Fuzzy system, Unfulfillment degrees, CMOS opamp.

ABSTRACT: The aim of the paper is to present a fuzzy optimization engine in the frame of the analogue circuit design. The optimzation engine uses fuzzy systems that compute the coefficient used to find new parameter values in each iteration. These fuzzy systems turn to good account the qualitative design knowledge and the designer experience. After introducing our fuzzy optimization method we design a basic two-stage CMOS operational amplifier.

I. INTRODUCTION

Mixed signal designs are increasing in number as a large fraction of new integrated circuits require an interface to the external, continuous-valued world. The digital portion of these designs can be attached with modern cell-based tools for synthesis, mapping, and physical design. The analog portion, however, is still routinely designed by hand. Although it is typically a small fraction of the overall design size (e.g. 10 000 – 20 000 analog transistors), the analog portion in this design is often the bottleneck because of the lack of automation tools [1].
The best way to rapidly design such mixed systems is to develop CAD tools than can automatically design analog cells [2].
Fig. 1 [1] illustrates the basic architectures of most analog synthesis tools. An *optimization engine* visits candidate circuit designs and modifies the values of the design parameters in such a way that the actual circuit performances meet the design requirements. An evaluation engine computes the actual circuit performances (based on the last parameter values), in order to compare them with the requirements.
In our paper we will focus only on the optimization engine. Finding a good optimization engine is not an easy task due to conflicting design requirements and performance constrains, which are generally implicit non-linear function of circuit parameters. In recent years, several prototype design automation systems have been

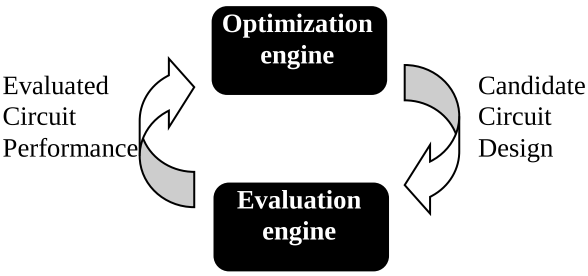

Fig. 1. Abstract model of analog synthesis tools

proposed to automate the design of cell-level analogue circuits. The optimisation methods used for analogue circuit design can be classified [3], [4] into four main categories:
1. Classical optimisation methods such as steepest descent, sequential quadratic programming, Lagrange multiplier method.
2. Global optimization methods such as depth first search, breadth first search, optimal solution search, branch and bound and simulated annealing.
3. Convex optimisation methods such as geometric programming.
4. Computational intelligence based methods such as genetic algorithm, evolutionary strategies, fuzzy logic, expert systems and neural networks. These methods usually find a locally optimal design or even just a *good* or *reasonable* design. The final design solution depends on the initial state and on the algorithm parameters.
In [5] and [6] a classical expert system is used for selecting a strategy to improve the performance. It uses heuristic rules that provide advice on how to modify {increase or decrease) some design parameters in order to obtain performances that meet the design requirements.
Our intention is to develop a strategy to compute the new values of circuit parameters, using the advantages offered by fuzzy sets and systems theory: degree of satisfaction of the predicate, accurate and simple acquisition of the designer knowledge, interpolation capabilities of the fuzzy systems, ability to deal with natural language and numerical data, etc.
This work proposes a fuzzy optimisation engine based on fuzzy systems that compute the new parameter values depending on the degree of unfulfillment of the design requirements.
The reminder of the paper is organised as follows. In section II we describe the proposed fuzzy optimization engine. In section III we offer some details about implementation of the method and some experimental results. Finally, in section IV, some conclusions are drawn.

II. THE FUZZY OPTIMIZATION ENGINE

The way (strategy) to compute new parameter values in the iterative optimization process (Fig. 1.) is a very important stage in the whole optimization procedure.

This strategy should be chosen so that the optimization will converge to an optimal solution in a reduced number of iterations. This task is not an easy one, due to the complex relations between design parameters and circuit performances. The same parameter can affect more than one circuit performance at a time, so that when a parameter is modified to improve a performance it can worsen another performance. Thus we should focus on finding the optimum change for each parameter value.

For example let's consider two performance functions:

$$f_1(x_1, x_2) = K_1 \frac{x_1}{\sqrt{x_2}} \quad (1)$$

$$f_2(x_1, x_2) = K_2 x_1^2 x_2 \quad (2)$$

where x_1 and x_2 are the design parameters and K_1, K_2 are constants.

For the above relations we can formulate some linguistic rules regarding x_1 parameter, based on the qualitative dependencies between design parameters and design requirements:

If f_1 must be increased **or** f_2 must be increased
then increase x_1.
If f_1 must be decreased **or** f_2 must be decreased
then decrease x_1.

The statement "f_1 must be increased" and "f_1 must be decreased", means that the actual value of f_1 is too small, respectively too big. So we can reformulate the above rules in the following manner:

If f_1 is smaller **or** f_2 is smaller
then increase x_1.
If f_1 is bigger **or** f_2 is bigger
then decrease x_1.

Our method proposes a fuzzy logic system for every design parameter, to compute a coefficient to modify the involved parameter. We use a zero order Takagi–Sugeno system with circuit performances as inputs and with coefficient as output (Fig. 2.).

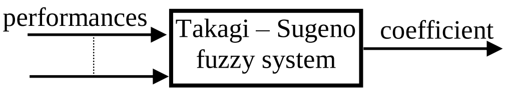

Fig. 2. Coefficient computing

The coefficient takes values in the range [-1; 1] and the corresponding parameters will be modified as follows:

$$\text{param} = \begin{cases} (1 + \eta_u \cdot \text{coefficient}) \cdot \text{param}; & \text{coefficient} \geq 0 \\ (1 + \eta_d \cdot \text{coefficient}) \cdot \text{param}; & \text{coefficient} < 0 \end{cases}$$

We use two dump factors to modify the parameter: η_u when the parameter value must go up and η_d when the parameter value must go down. The values of these parameters can be choose by the user, their default values being η_u =1 and η_d =0.5.

In order to describe the fuzzy systems we will use the notations:

$x=[x_1, \ldots, x_M]$ – the vector of design parameters;

$f_1(x)$, …, $f_i(x)$, …, $f_N(x)$ – the circuit performance functions;

r_1 ,…, r_i , … , r_N – the numerical values of the design requirements.

Every design requirement, can be stated as follows: $f_i(x)= r_i$, or $f_i(x)> r_i$, or $f_i(x)< r_i$.

The fuzzy sets for the inputs (performances) are built as follows:

For the " $f_i(x)= r_i$ " requirements we use the fuzzy sets presented in Fig. 3. a)., and for "$f_i(x)> r_i$ " requirements we use the fuzzy set presented in Fig. 3. b).

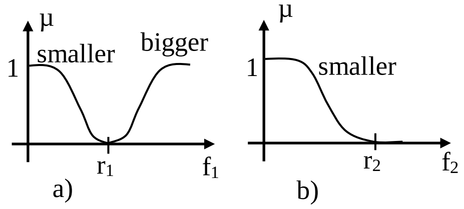

Fig. 3. Fuzzy sets for:
a) $f_1(x)=r_1$; b) $f_2(x)>r_2$

This way we have a measure of the unfulfillment degree of the requirements; $\mu_{r_i}(f_i(x))$ indicating the error in accomplish the i[th] design requirements.

For every output linguistic variable we use singleton fuzzy sets (Fig . 4)

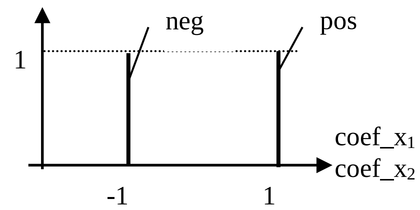

Fig. 4. Output fuzzy sets

For example using the $f_1(x_1,x_2)$ and $f_2(x_1,x_2)$ circuit function mentioned above, the base rule for the coefficient of x_1, coef_x_1, is:

R1*: **If** f_1 is smaller or f_2 is smaller **then** coef_x_1 is pos*

R2*: **If** f_1 is bigger **then** coef_x_1 is neg*

The inference process for coef_x_1 is presented in Fig. 5., where f_1^* and f_2^* are the current values for performances and coef*_x_1 is the resulting values for coef_x_1.

We can see that the value of coef_x_1 is a weighted sum of the two values resulting from the rules, the weights being the activation degree μ of each rule, or in other words the error in accomplishing the requirements.

Our fuzzy systems use 'max' operator for *or* method, 'min' operator for *implication* method and 'wtsum' operator for *defuzzification* method. Due to 'max' for *or* method the optimization engine acts, in each iteration, in order to reduce the biggest error.

Also we should mention here that the 'wtsum' for *defuzzification* make the optimization engine to act in an adaptive manner: the modification of a parameter will be smaller for a small error and bigger for a big error. For the sake of simplicity let's suppose that performance f_1 is fully satisfied (μ_2=0). If f_2 is not satisfied at all, meaning μ_1=1, the value of x_1 will be 100% increased. If f_2 is almost satisfied meaning, let's say μ_1=0.15, the value of x_1 will be only 15% increase.

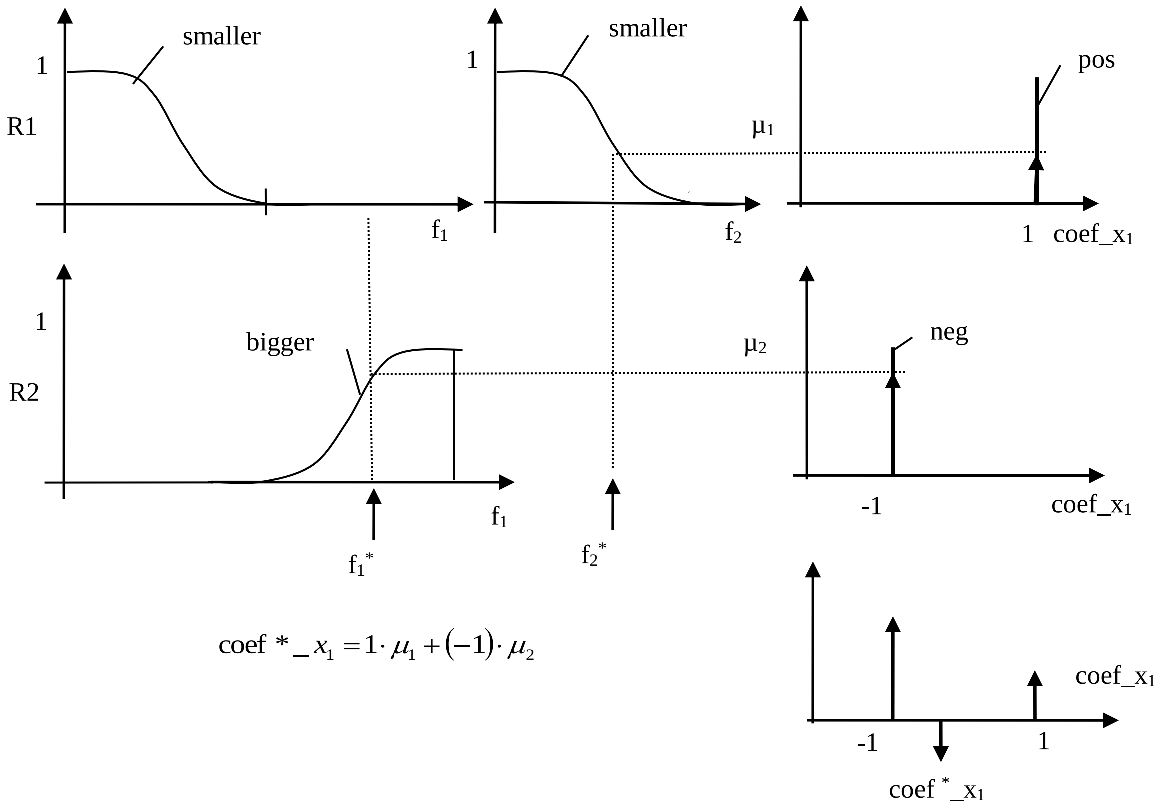

Fig. 5. Inference process for $coef\_x_1$

III. IMPLEMENTATION AND RESULTS

Using our fuzzy optimization engine we designed a two stage CMOS op-amp presented in Fig. 6. We selected this circuits because it is used in almost all the paper dealing with design automation of analogue electronic circuits [1] [3], [7], [8], [9], [10], [11]. As independent design parameters we consider:

- (W/L)1 for T1;
- (W/L)6 for T6;
- ib for bias current IB;
- cc for compensation capacitor Cc.

We consider fixed values for two parameters (W/L)5 = 10 for T5 and (W/L)7 =1, for T7. The others design parameter results from the following relations:

$$(W/L)2 = (W/L)1; \quad (3)$$

$$(W/L)8 = (W/L)7; \quad (4)$$

$$(W/L)3 = (W/L)4 = \frac{1}{2}\frac{(W/L)6 \cdot (W/L)7}{(W/L)5} \quad (5)$$

Also we use V_{DD}=+2.5V, V_{SS}=-2.5V and CL=10pF. We consider the following circuit performances: DC gain (avo), Gain Bandwidth (gbw), Phase Margin (pm) and

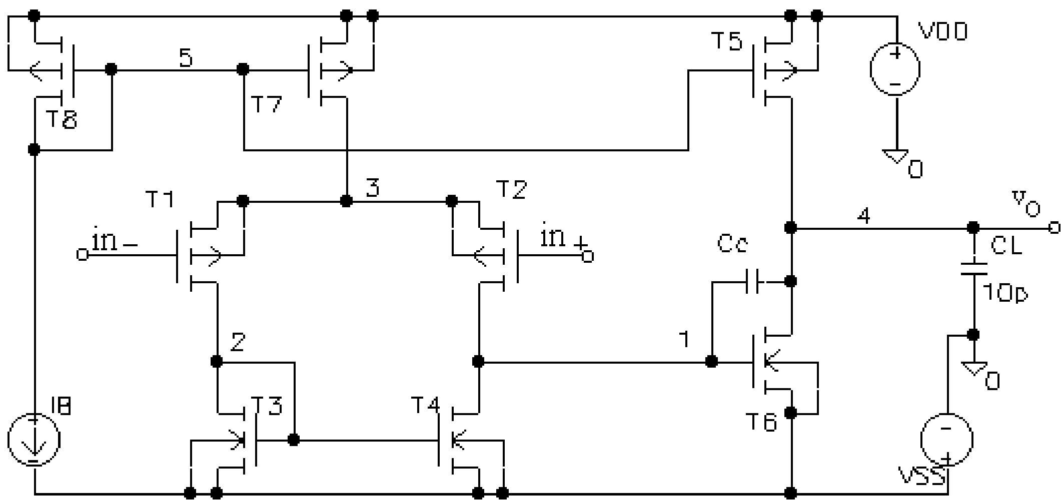

Fig. 6. Basic two stage CMOS operational amplifier

Slew Rate (sr)

We implemented the entire optimization algorithm (Fig.1) in Matlab. The performance is evaluated using fuzzy systems that model the circuit performance functions.

As the error criterion we use the maximum of the error degrees in accomplishing the design requirements (see previous section). The fuzzy sets of the input variables and the base rules of the fuzzy system that compute the coefficient for computing new parameter values are built automatically, depending on the design requirements. For example for the requirements:

avo = 450 000 [V/V];
gbw ≥ 1500 [KHz];
sr ≥ 1.5 [V/µs];
pm ≥ 60 [º]

the fuzzy system for coef_ib (the coefficient to modify ib parameter) has the input fuzzy sets represented in Fig. 7.

The fuzzy sets for the output are similar with the ones in Fig. 5., but the name of the output variable is 'coef_ib'. The rule base contains the following rules:

1. If (avo is smaller) then (coef_ib is neg) (1)
2. If (avo is bigger) or (gbw is smaller) or (sr is smaller) then (coef_ib is pos) (1)

The design requirements, the final performances and the final errors are presented in Table 1. We mention that in our implementation we can repeat the whole optimization algorithm with different initial values of the design parameters, randomly chosen. We do this in order to increase the chance to find a solution close to the global optimum. Naturally we choose as the final solution the one with minimum error (zero if possible). The evolution of the maximum error in each iteration is presented in Fig. 8.

TABLE 1.

	requirements	performances	errors
avo [V/V]	= 450000	445438.65	0.00022
gbw [KHz]	≥ 1500	1527.13	0
pm [º]	≥ 60	60.58	0
sr [V/µs]	≥ 1.5	1.51	0

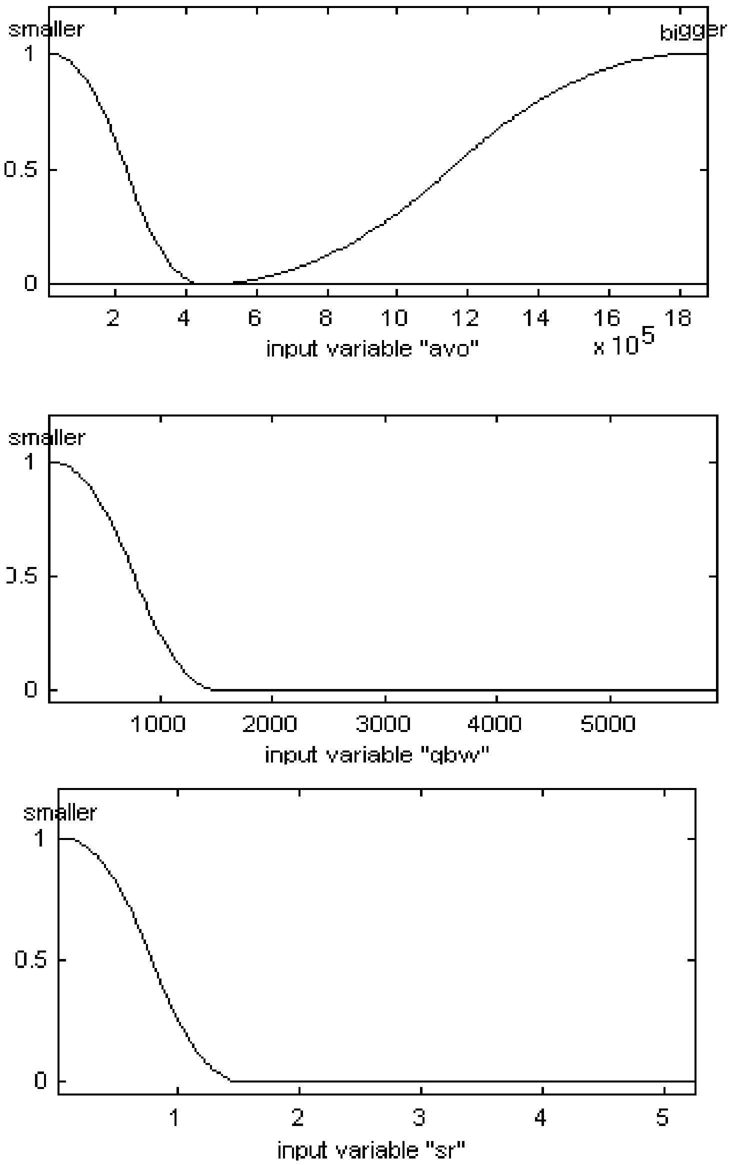

Fig. 7. The input fuzzy sets for coef_ib

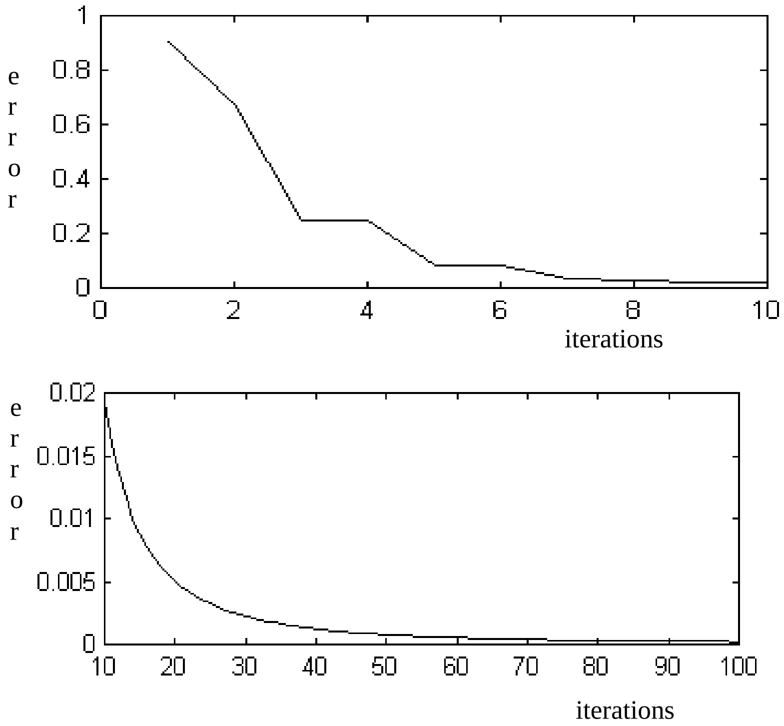

Fig. 8. The evolution of the maximum error

As experimental results show, the requirements for gbw, pm and sr circuit function are fully satisfied. Also we find a reasonable value for avo requirements, the error being very small, 0.00022. We notice that for avo there is an 'equal' requirement, stronger than the 'greater or egual' requirements as for the others circuit functions.

In keeping with Fig.8. the design is rapidly improved during the first iterations. After that, when we are in the proximity of the solution, the design is slowly improved (fine-tuning). Also the Fig. 8. gives us an image of the rapid convergence of the fuzzy optimization engine.

IV. CONCLUSION

In this paper we have proposed a new optimization engine, based on fuzzy logic system, for designing analogue circuits. A fuzzy system contributes to determine the new value for each parameter, depending on the unfulfillment degrees of all the requirements. In order to validate our method we implemented it in Matlab and used it to design a basic two stage CMOS operational amplifier. The results confirm a rapid convergence of the optimization engine. Since the designing of the circuit was successfully, it is necessary to try to design circuits of larger scale. In such designs our optimization engine should be more effective.

THE AUTHORS

Assist.Prof. Gabriel Oltean, Assoc.Prof Sorin Hintea and Assist. Doris Lupea are with the Department of Basis of Electronics, Technical University of Cluj-Napoca, Str C. Daicoviciu No 15, Romania.
E-mail: Gabriel.Oltean@bel.utcluj.ro

REFERENCES

[1] Phelps, R., Krasnicki, M., Rutenbar, R.A., Carley, R.L., Hellums, J.R., "Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search". IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 19, No.6., June 2000, pp. 703-717.

[2] Fares, M., Kaminska, Bozena, "FPAD: A Fuzzy Nonlinear Programming Approach to Analog Circuit Design", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 14, no.7, July 1995, pp. 785-793;

[3] Hershenson, Maria del Mar, Boyd S.P., Lee T.H., "GPCAD: A Tool for CMOS Op- Amp Synthesis", International Conference on CAD '98;

[4] Oltean, G., "Applications Possibility of the Fuzzy Mathematics to Analyse and Design some Electronic Circuits", Ph.D. Report, Technical University of Cluj-Napoca, Romania, 1999;

[5] Sheu, B.J., Fung, A.H., Lai, Y-H.,"A Knowledge-Based Approach to Analog IC Design", IEEE Transactions on Circuits and Systems, vol. 35, no.2, February 1988, pp256-258;

[6] Hashizume, M., Kawai, H.Y., Nii,K., Tamesada,T., "Design Automation System for Analog Circuits Based on Fuzzy Logic", IEEE Custom Integrated Circuits Conference, 1989 pp. 461-464;

[7] Prakobwaitayakit,K., Fujii,N.," Circuit Topology Selection Based on Neural Network", IEEE, 1998, pp.387-390;

[8] Torralba, A., Chavez, J., Franquelo, L.G., "FASY: A Fuzzy-Logic Based Tool for Analog Syhesis", IEEE Trans. Computer Aided Design, Vol.15, No7, July1996, pp.705-715;

[9] Torralba, A., Chavez, J., Franquelo, L.G., "A collection of Fuzzy Logic-Based Tools for the Automated Design, Modelling and Test of Analog Circuits", IEEE Micro, Vol 16, No 4, Aug 1996, pp 61-68;
[10] Swings, K., Sansen, W., "ARIADNE: A Constraint-Based Approach to Computer-Aided Synthesis and Modeling of Analog Integrated Circuits", Analog Integrated Circuits and Signal Processing 3, 197-215 , Kluwer Academic Publishers, Boston. 1993, pp. 37-55;
[11] Rodriguez, B., R., S., Styblinski, M., A., "Adaptive Hierarchical Multi-Ojective Fuzzy Optimization for Circuit Design", IEEE, 1993, pp. 1813-1816;