Volume 42, Number 1, 2002

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

ANALOG DESIGN: MULTIOBJECTIVE OPTIMIZATION METHOD BASED ON FUZZY LOGIC

Gabriel OLTEAN, Costin MIRON
Bases of Electronics Dept. Technical University of Cluj-Napoca, Romania
C.Daicoviciu nr.15, tel 195699 ext 186, email:Gabriel.Oltean@bel.utcluj.ro

Abstract: The paper presents a new method for optimizing the analog circuits design. The method performs a multiobjective optimization, the parameter modification taking into account the unfulfillement degrees of all the requirements. The method uses fuzzy sets to define fuzzy objectives and fuzzy systems to compute new parameter values. The solution of a multiobjective optimization is a set of Pareto optimal points. We can obtain the set of Pareto optimal points using the idea of population of solutions. We select as the final optimal solution the one with smallest mean of unfulfillement degrees of requirements (zero if possible). The strategy to compute new parameter values uses local gradient information and encapsulates human expert thinking: to improve a performance we modify mostly the parameter with more influence. This way every Pareto optimal point can be found with accuracy.
After introducing our optimization method, we optimize the design of a common-emitter amplifier. As expected the results prove that the method is efficient. For different sets of requirements we found a set of Pareto optimal points and for the sets with non-conflicting requirements the final solution has the mean of unfulfillement degrees equal to zero.

Keywords: *design, fuzzy optimization, multiobjective, Pareto set, population of solutions*
.

ANALOG DESIGN: MULTIOBJECTIVE OPTIMIZATION METHOD BASED ON FUZZY LOGIC

Gabriel OLTEAN, Costin MIRON
Bases of Electronics Dept. Technical University of Cluj-Napoca, Romania
C.Daicoviciu nr.15, tel 195699 ext 186, email:Gabriel.Oltean@bel.utcluj.ro

Abstract: The paper presents a new method for optimizing the analog circuits design. The method performs a multiobjective optimization, the parameter modification taking into account the unfulfillement degrees of all the requirements. The method uses fuzzy sets to define fuzzy objectives and fuzzy systems to compute new parameter values. The solution of a multiobjective optimization is a set of Pareto optimal points. We can obtain the set of Pareto optimal points using the idea of population of solutions. We select as the final optimal solution the one with smallest mean of unfulfillement degrees of requirements (zero if possible). The strategy to compute new parameter values uses local gradient information and encapsulates human expert thinking: to improve a performance we modify mostly the parameter with more influence. This way every Pareto optimal point can be found with accuracy.
After introducing our optimization method, we optimize the design of a common-emitter amplifier. As expected the results prove that the method is efficient. For different sets of requirements we found a set of Pareto optimal points and for the sets with non-conflicting requirements the final solution has the mean of unfulfillement degrees equal to zero.

Keywords: *design, fuzzy optimization, multiobjective, Pareto set, population of solutions*

I. INTRODUCTION

Even we are in the digital age, everyone is finding the need for a little bit of analog, at least at the interface between the electronic systems and the “real” world. According to [1] nearly 70 percent of all ICs will have analog components within five years, compared with about 25 percent today.

As the demand for mixed-mode integrated circuits increases, the design of analog circuit becomes more critical. Many authors have noted the disproportionately large design time devoted to the analog circuitry in mixed mode integrated circuits. [2] Predictions abound that analog design will become a barrier for the next generation system-on-a-chip design [3].

The best way to rapidly design such mixed system is to develop CAD tools that can automatically design analog cells [4].

As Gielen and Rutenbar showed [5], the task of circuit sizing is placed at the lowest level of design hierarchy, where the design parameters (sizes and biasing off all devices) have to be determined so that the circuit performances meet the specified design requirements. This mapping from design requirements into proper, preferably optimal device size and biasing, in general involves solving the set of physical equations that relate the device sizes to the electrical circuit performances.

Solving these equations explicitly is in general not possible so two basic alternatives are used: the knowledge-based approach and the optimization-based approach. In our paper we shall focus only on the latter approach optimization-based approach the circuit performances are written as objective functions to be maximized (minimized). So we face a multiobjective optimization problem (MOP). Solution of multiobjective problem are known as noninferior or Pareto-optimal points [6], [7].

A global noninferior or Pareto-optimal point is, generally speaking, a point x in the feasible space so that there is no other point x′ in that space with superior values for all the objective functions (in the case of objective maximization). The definition of a local noninferior point is similar, except that it would consider a neighborhood of x (for a rigorous mathematical definition, see [7]). Any method for solving a MOP must be able to generate a set of noninferior solutions [6].

One way to solve the MOP is the Goal Attainment method of Gembicki that uses a set of design goals associated with the set of objective functions [7]. An alternate procedure for dealing with multiobjective optimization is to simultaneously optimize all the objectives.

The idea of using fuzzy techniques in analog design optimization can be met in some previous papers. In [4] and [8] fuzzy sets are used to represent fuzzy meanings of the design requirements and constrains. This way crisp objectives are transformed into fuzzy.The membership function is regarded objectives as a degree of fulfillment of the associated fuzzy objectives. Then the optimization problem consists in maximizing a weighted sum of these degrees of fulfillment.

In [9] fuzzy sets are used as an error measure between the design requirements and circuit performances. This error measure is called unfulfillment degree of requirement (UDR) and the optimization problem consist in minimizing each of these UDRs. Moreover the optimization method itself uses fuzzy systems to find the new parameters value in each iteration. For every design parameter a zero order Takagi-Sugeno fuzzy system (that incorporate the performances-parameters global qualitative dependencies) compute a coefficient to modify it, depending on all UDRs. Two drawbacks affect this approach: first, the performance function should be monotonous and second, the method can find only a local noninferior solution instead of a global one.

A new multiobjective optimization method based on fuzzy logic is proposed in this paper. The new method uses a populations of solutions to find a set of local Pareto optimal points from which we can choose the most suitable one; this way we increase very much the probability of finding the global Pareto optimal point. In order to accurately find the local Pareto optimal points we use quantitative information of the local gradient to compute new parameter values. The design requirements are fuzzified using fuzzy sets, so that the optimization problem has fuzzy objectives. We will use the unfulfillment degrees of the requirements (UDRs) as a measure of the objective achievement. Also, the strategy to compute new parameter values benefits by one of the advantages offered by fuzzy logic: accurate and simple acquisition of the human expert knowledge and thinking.

The reminder of this paper is organized as follows: Section 2 describes the proposed optimization method with population of solutions idea and the strategy to compute new parameter values. Section 3 briefly presents the implementation of the method and the results obtained by optimizing the design of a basic BJT amplifier. Finally section 4 presents some conclusions.

II. THE FUZZY OPTIMIZATION METHOD

The optimization method should be chosen so that it converges to a global optimal solution in a reduced number of iterations. This is not a simple task due to the complex relations between design parameters and circuit performances. A parameter affects more than one circuit performance at the same time so when it is modified in order to improve a performance it can damage another.

A. The idea of population of solutions

Starting the optimization with only one initial solution, we can remain blocked into a local Pareto optimal point, where an improvement in one objective requires a degradation of another. If we can obtain a set of local Pareto optimal points it is highly possible to have the global Pareto optimal point among them.

So, instead of using one search path we suggest using a parallel search dealing with the idea of population of solutions consisting of candidate solutions. The optimization starts with the initial candidate solutions. These initial candidate solutions can be obtained in several ways: randomly generated, computed with approximate design equations, etc.

At each iteration, for every candidate solution the performances, the UDRs and new parameter values are computed. If the UDRs for one candidate can not be decreased anymore, we have found a local Pareto optimal point and the future iterations will not visit this candidate solution, shortening the entire optimization time.

The optimization algorithm stops in one of the following situations:

i) all the UDRs become zero for one candidate solution. This candidate solution is considered a global Pareto optimal point and it is our final optimal solution. We will not continue to search other Pareto optimal point on the remaining search paths.

ii) none of the candidate solutions can be further improved, meaning that the set of local Pareto optimal points was obtained.

As the final optimal solution we chose the one with the minimum value of the mean of unfulfillment degrees of requirements (MUDR), considered as global optimal point.

B. New parameter values computing

The method for computing the new values for the design parameters involves fuzzy techniques and local gradient information.

Generally speaking each design parameter can affect more or less each circuit performance. In our method the sign and the value to modify a certain parameter takes into account the UDRs, the gradients and the relative importance (weight) of the involved parameters in relations with the circuit performances.

Our method acts as a human expert for a certain circuit performance:

- it is better to modify more the parameter with greater weight, because it can really affect the performance, and the modification also depends on the unfulfillment degrees of the corresponding requirements.
- the parameter with lower weight is modified less or not at all, because its influence on circuit performance is insignificant.
- the final modification of a parameter is a weighted sum of the partial modification (imposed by every performance function).

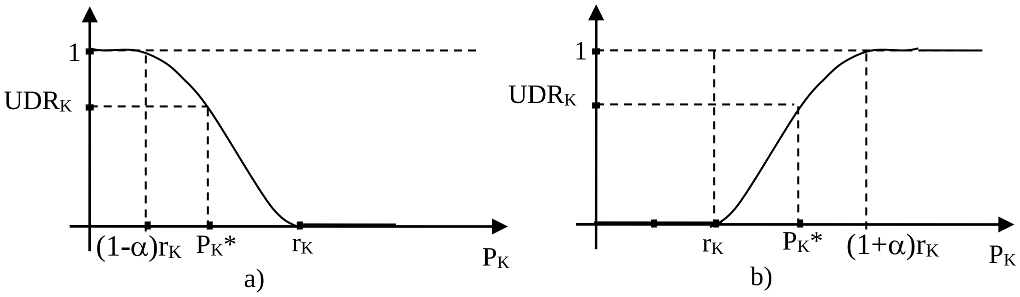

Fig.1 Computing UDR$_K$ for fuzzy objective:
a) "P$_K$ greater or equal with r$_K$ "; b) "P$_K$ less or equal with r$_K$ "

To explain the way new parameter values are computed let us consider the notations:

Such human expert knowledge is captured and incorporated in our method by means of a fuzzy logic system.

P_k – the k[th] performance function, $k = \overline{1, N}$

x_i – the i[th] design parameter, $i = \overline{1, M}$

In each iteration:

i) Compute the local gradients of the performances in relation with each design parameter
$\nabla P_k(x_i)$ - the local gradient of P_k performance in relation with x_i parameter.

ii) For every performance P_k we compute the weights $w_{P_k}(x_i)$ that shows the relative importance of every x_i in modifying the performance P_k. These weights are computed based on absolute values of the local gradients:

$$w_{P_k}(x_i) = \frac{|\nabla P_k(x_i)|}{\sum_{i=1}^{M} |\nabla P_k(x_i)|}; i = \overline{1, M}; k = \overline{1, N} \qquad (1)$$

iii) For every circuit requirements r_k compute the UDR_k as a membership degree of the actual value of the corresponding performances P_k^* to the associated fuzzy objectives.An example is shown in Fig.1.a) for with r_k". A value UDR=0 means a fully achievement of the fuzzy objectives, while UDR=1 means the fuzzy objective is not achieve at all. α is a factor in (0;1) domain that can control the UDR via the shape of the fuzzy sets that define fuzzy objectives. the fuzzy objective "P_k greater or equal with r_k" and in Fig.1.b.) for the fuzzy objective "P_k less or equal

iv) For every parameter x_i and every performance P_k we compute a partial coefficient to modify the parameter. This partial coefficient $coef_{xi}(P_k)$ is computed by first order Takagi-Sugeno fuzzy system (Fig. 2.)

The fuzzy sets for the input linguistic variables "UDR" and "weight" are presented in Fig.3. and for the output linguistic variable part_coeff in Fig.4.

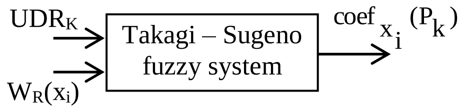

Fig.2 Partial coefficient computing

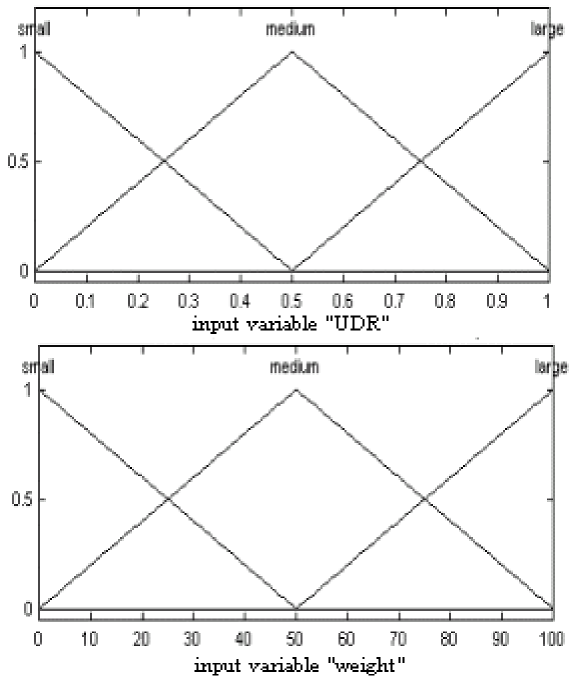

Fig.3 The fuzzy sets for the inputs

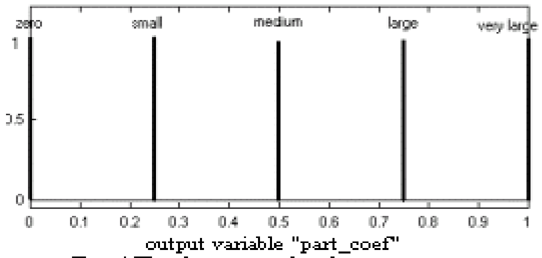

Fig.4 The fuzzy sets for the output

The fuzzy rules are presented in Table 1. where, for example the 4[th] column and the 2[nd] row give the following fuzzy rules:

"If UDR is Large and weight is Small then part_coeff is Medium".

Table 1. The fuzzy rules

weight \ UDR	S	M	L
S	Z	S	M
M	S	M	L
L	S	L	VL

Z - Zero
S - Small
M - Medium
L - Large
VL - Very Large

v) The partial coefficients $coef_{x_i}(P_k)$ receive a plus or minus sign depending on the sign of the local gradient and on the direction (go up or go down) in which the performance must be modified. So we have obtained partial coefficients that have a sign: $scoef_{x_i}(P_k)$.

vi) For every parameter x_i we compute the weights $w_{x_i}(P_k)$ that show the relative importance of every performance P_k to compute the modification of x_i parameter.

$$w_{x_i}(P_k) = \frac{|\nabla P_k(x_i)|}{\sum_{k=1}^{N} |\nabla P_k(x_i)|}; k = \overline{1,N}; i = \overline{1,M} \qquad (2)$$

vii) The coefficients used for modifying each parameter are computed as weighted sum of the partial coefficients, the weight being $w_{x_i}(P_k)$. It means that the greater the weight is, the greater the influence on the partial coefficient.

$$scoef_{x_i} = \frac{\sum_{k=1}^{N} \left(w_{x_i}(P_k) \cdot scoef_{x_i}(P_k) \right)}{\sum_{k=1}^{N} \left| w_{x_i}(P_k) \right|} \qquad (3)$$

viii) Compute new parameter values of the design parameters:

$$x_i = \left(1 + scoef_{x_i}\right) \cdot x_i \qquad (4)$$

Finally we should mention that the optimization method acts in an adaptive manner: when the UDRs are large (towards 1) we have large coefficients to modify the parameters (see Table 1). For small UDRs we have small coefficients to modify the parameters, so we can focus our search so that the solution converges to the exact local Pareto optimal point.

III. IMPLEMENTATION AND RESULTS

In order to evaluate our multiobjective optimization algoritm we implemented a prototype system in Matlab for Windows. The prototype consist on a main function "optfuzz" and other secondary function. The main function should be invoke from Matlab workspace with a series of arguments: optfuzz (`fun`, reqs, sign, ub, lb).

- fun – a string containing the name of the Matlab function that computes the objective functions;
- reqs – vector of numerical values of the requirements;
- sign – vector with +1 or –1 values, with the same length as reqs vector. When the values is +1 "optfuzz" attempt to make the objective function greater or equal to corresponding requirements. When the value is –1 "optfuzz" attempt to make the objective function less than the corresponding requirements;
- ub – a vector of upper bounds of the parameters;
- lb – a vector of lower bounds of the parameters.

Also, we can set some options: number of iterations, number of candidate solutions and α factor (see Fig.1). The initial values of the parameters are randomly generated for each candidate solution.

The user should only write his objective functions and run the "optfuzz" with the arguments show above. The optimization routine return the final values of objective function, the values of the parameters, the UDR for each requirements and a curve with the evolution of MUDR during the optimization for the candidate solution that provide final solution.

Using our multiobjective optimization based on fuzzy logic we designed a common-emitter amplifier presented in Fig. 5.

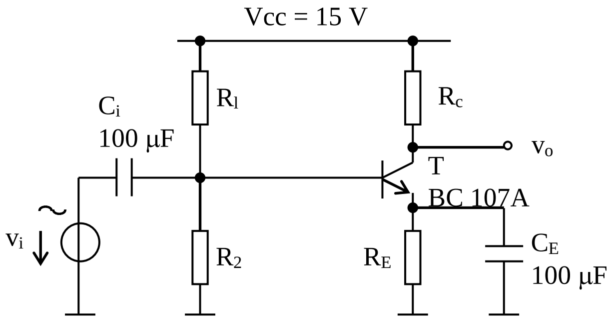

Fig. 5. Common-emitter amplifier

Table 2.

Require ments (set I)	Candidate solutions									
	1°		2°		3°		4°		5°	
	Perf.	UDR	Perf.	UDR	Perf.	UDR	Perf.	UDR	Perf.	UDR
Ri ≥ 2 kΩ	1.829	0.091	1.822	0.099	1.823	0.097	1.798	0.127	1.822	0.099
Ro<1.2 kΩ	1.321	0.126	1.316	0.118	1.318	0.121	1.302	0.09	1.318	0.121
Av ≥ 150	139.34	0.0002	149.5	0.0001	149.53	0.001	149.75	0.00004	149.65	0.00007
B ≥20 kHz	42.28	0.0	42.36	0.0	42.32	0.0	42.66	0.0	42.31	0.0
		0.0544		0.0542		0.0545		0.0544		0.0549

Table 3.

Parameters	Candidate solutions				
	1°	2°	3°	4°	5°
R_2 [kΩ]	67.522	66.689	66.791	62.103	66.369
R_E [kΩ]	2.159	2.134	2.138	2.021	2.128
R_C [kΩ]	1.366	1.362	1.363	1.347	1.363

Table 4

	set II			set III		
	Requir	Perform	MUDR	Requir	Perform	MUDR
R_i [kΩ]	>1.8	1.805	0	>2.5	2.502	0
R_o [kΩ]	<0.7	0.621	0	<0.15	0.126	0
Av	>70	71.324	0	>9.8	9.831	0
B [kHz]	>90	91.058	0	>500	504.139	0

We consider four performance functions: input resistance R_i; output resistance R_o; passband gain Av; bandwidth B. As design parameter we consider three resistors: R_2, R_E and R_C. The dependencies of performance functions on design parameters are expressed by analytical equations.

We try to optimize the design of the amplifier for the set of requirements (set I) presented in Table 2. The optimization was run for a population of five candidate solutions, namely 1°…5°. The results: final performances, UDR and MUDR are also presented in Table 2.

The final parameter values for each candidate solutions form a set of local Pareto optimal points and they are presented in Table 3.

As the final solution (global Pareto optimal points) we choose the one with the smallest MUDR. So, our final solution is the one provided by candidate solution 2° withMUDR=0.0542. For this solution the requirement for B is fully realized, for Av almost realized while for Ri and Ro are less realized. This means that our requirements are competitive and cannot be fully achieved concurrently

We optimize the design for another two sets, set II and set III, of design requirements presented in Table 4. Also in Table 4 we present the final performances and MUDR for both sets of requirements. The design optimization is successfully, all the requirements being fully attained. In both cases we use a population of 5 candidate solution. The evolution of the MUDR for all five candidate solutions from set II are presented in Fig. 6.

We can see that the MUDR are improved during optimization, for all candidate solutions. The optimization is stopped when the MUDR for one candidate solution reaches the zero value, means that all the requirements are fully achieved. This candidate solution (4) gives the final solution. Fig. 7 and Fig. 8 show the evolutions of MUDRs for candidate solutions that give final solutions.

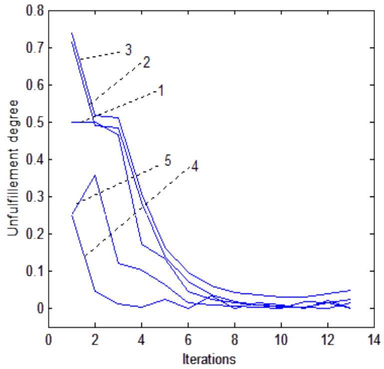

Fig. 6. Evolution of MUDR

In Table 5 we present the initial and final values of the design parameters for candidate solutions that gives the final solution in set II and set III.

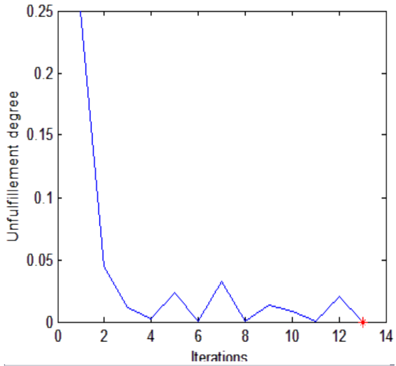

Fig. 7. Evolution of MUDR in set II.

Table 5

Parameters	set II		set III	
	initial	final	initial	final
R_2 [kΩ]	61.846	67.618	38.451	55.215
R_E [kΩ]	2.792	2.125	3.087	2.858
R_C [kΩ]	0.483	0.631	0.411	0.127

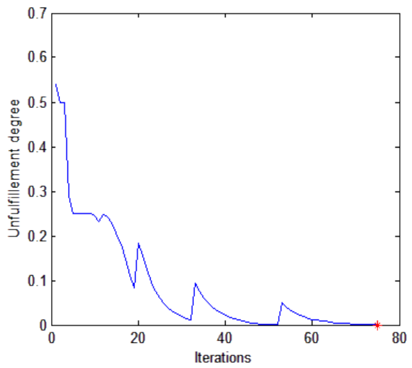

Fig. 8. Evolution of MUDR in set III

For a better appreciation of our fuzzy multiobjective optimization method we compare our results with the ones obtained using another multiobjective optimization method: Goal Attainment method of Gembicki from the optimization Toolbox of Matlab "fgoalattain" [10]. Since this method requires an initial starting points we use the same starting point as for the "optfuzz" presented in Table 5. The results obtained with "fgoalattain" are show in Table 6.

Table 6

	set II			set III		
	Requir	Perform	MUDR	Requir	Perform	MUDR
R_i [kΩ]	>1.8	1.841	0	>2.5	2.480	0.00079
R_o [kΩ]	<0.7	0.684	0	<0.15	0.151	0.00079
Av	>70	71.609	0	> 9.8	9.771	0.00079
B [kHz]	>90	85.034	0.038	>500	437.332	0.19636

We should mention that we use the same performance functions, the same Matlab version and the same computer. Because the "fgoalattain" do not compute UDR, we made those calculations for final performances. Some compative results can be seen in Table 7.

Table 7

	set II		set III	
Optimization method	optfuzz	fgoalattain	optfuzz	fgoalattain
MUDR	0	0.0095	0	0.0496
Iterations	13	66	75	113
CPU time [s]	2.06	3.08	8.73	6.11

We can see that our method find better final solutions (MUDR=0) than "fgoalattain" (MUDR=0.0095 or 0.0496). Also, the number of iterations is smaller for our method while the and the CPU time is comparable for the two methods. The "optfuzz" method also has the advantages of a very large chance to find the global optimal solution compared with "fgoalattain" that can be trapped into a local solution.

IV. CONCLUSION

In this paper a new multiobjective optimization method for analog circuits design using fuzzy logic has been introduced. Fuzzy sets were used to define fuzzy objectives and a fuzzy system that incorporates human thinking contributes to compute new parameters values.

The method really allows optimization of several objectives simultaneously because the modification of the parameters is a function of the unfulfillment degrees of all the requirements.

The results obtained after optimizing the design of a common-emitter amplifier show that our method works very well. Due to the population of solutions we find a set of local Pareto optimal points and the point with minimum MUDR is chosed as a final optimal solution. The method has a very large chance to find the global optimal solution due to its multiple search path. Also in the proximity of the final solutions the method works well to continue decrease MUDR up to the local Pareto optimal points. The quality of each final solution is very high. This is possible because the method uses local gradient information and works in an adaptive manner: while the UDR decrease, the step in the parameter modification also decreases.

Compared with other multiobjective optimization method ("fgoalattain") our method prove to be superior in some aspects: better quality of final solutions, smaller number of iterations; higher chance to find a global solution.

REFERENCES

[1] B. Martin, "Automation Comes to Analog", *IEEE Spectrum*, Vol. 38, No. 6, pp. 70-75, June 2001.

[2] M. D. M. Hershenson, S.P. Boyd and T.H. Lee, "Optimal Design of a CMOS Op-amp via Geometric Programming", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and System*, Vol. 20, No.1, pp. 1-21, January 2001.

[3] S. Boyd, "New Approaches Speed up Optimization of Analog Design", *Electronic Design*, 48(20):62, October 2000.

[4] M. Fares and B. Kaminska, "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, pp. 785-793, July 1995.

[5] G. G. E. Gielen and R. A. Rutenbar, "Computer Aided Design of Analog and Mixed Signal Integrated Circuits", *Proceedings of IEEE*, Vol.88, No.12, pp 1825-1852, Dec. 2000.

[6] P. C. Maulik, "Comments on "FPAD: A Fuzzy Nonlinear Programming Approach to Analog Circuit Design"", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, No.6, pp. 656, June 1997.

[7] M.A. Branch and A. Grace, "*Optimization Toolbox User's Guide. For Use with MATLAB*", The MathWorks, Inc., 1996.

[8] B.R.S., Rodrigues, M.A., Styblinski, "Adaptive Hierarhical Multi-Objective Fuzzy Optimization for Circuits Design", *IEEE*, 1993, pp. 1813-1816.

[9] G., Oltean, C., Miron, Sorina, Zahan, Mihaela, Gordan, "A Fuzzy Optimization Method for CMOS Operational Amplifier Design", *Proc. of the 5th Seminar on Neural Network Applications in Electrical Engineering, IEEE, NEUREL2000,* Belgrade, Yugloslavia, 25-27 September, 2000, pp. 152-157

[10] ****, Optimization Toolbox Help for Matlab R12, The MathWorks, Inc., 2000.