Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 593684, 12 pages
http://dx.doi.org/10.1155/2014/593684

Research Article

Analog Circuit Design Optimization Based on Evolutionary Algorithms

Mansour Barari,[1] Hamid Reza Karimi,[2] and Farhad Razaghian[1]

[1] *Electrical Engineering Department, Islamic Azad University, South Tehran Branch, Tehran 11365/4435, Iran*
[2] *Department of Engineering, Faculty of Engineering and Science, University of Agder, 4898 Grimstad, Norway*

Correspondence should be addressed to Hamid Reza Karimi; hamid.r.karimi@uia.no

Received 3 January 2014; Accepted 28 February 2014; Published 7 April 2014

Academic Editor: Yang Tang

Copyright © 2014 Mansour Barari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates an evolutionary-based designing system for automated sizing of analog integrated circuits (ICs). Two evolutionary algorithms, genetic algorithm and PSO (Parswal particle swarm optimization) algorithm, are proposed to design analog ICs with practical user-defined specifications. On the basis of the combination of HSPICE and MATLAB, the system links circuit performances, evaluated through specific electrical simulation, to the optimization system in the MATLAB environment, for the selected topology. The system has been tested by typical and hard-to-design cases, such as complex analog blocks with stringent design requirements. The results show that the design specifications are closely met. Comparisons with available methods like genetic algorithms show that the proposed algorithm offers important advantages in terms of optimization quality and robustness. Moreover, the algorithm is shown to be efficient.

1. Introduction

Today, with the development of VLSI technology in order to integrate digital and analogue circuits as a complete system on a chip, the issue of optimal electronic circuits design, due to its abundant advantages, in contrast with manual design is very important. In this regard, because of their design complexity and difficulty, analogue circuits have attracted greater attention in optimization. In general,the analogue circuits design is undertaken in two stages. The first stage is related to the selection of the circuit structure so that it would have the necessary abilities to fulfill our expectations from the circuit. The second stage, which is often more time-consuming, is the optimization of the circuit parameters, for example, the size of the transistors and the value of the elements, so that it could satisfy output parameters, for example, consuming power and gain. Moreover, issues, depending on the nature of the main function and limitations (e.g., whether the fitness function is convex or not, limits are linear), can be solved by KKT (Karush-Kuhn-Tucker) condition theories, Lagrange coefficients, nonlinear/linear complementary problem (NLCP/LCP), and nonaffine/affine variational inequality (NAVI/AVI). However, due to its complexity, being time-consuming and case-dependent, there was a need for newer and more comprehensive solutions. For these reasons and in order to respond to these problems, the use of evolutionary optimization methods, such as genetic algorithm, particle swarm optimization, ant colony, and harmonic search algorithm, was proposed.

All these methods obey the following issues:

$$\begin{aligned} \min_{\underline{x}} \quad & f(\underline{x}) \\ \text{subject to} \quad & h(\underline{x}) = 0 \\ & g(\underline{x}) \le 0 \\ & \underline{X}_L \le \underline{x} \le \underline{X}_H. \end{aligned} \tag{1}$$

In this optimization, $f(\underline{x})$ is the function and must be minimized and $h(\underline{x})$, $g(\underline{x})$ are boundary conditions. The equality in (1) is our fitness function and refers to Kirchhoff laws (KVL and KCL).

Vector $\underline{x}$ is the design variable and $\underline{X}_L$ and $\underline{X}_H$ are their lower and upper bounds, respectively. Analog integrated circuits use (1) as a constrained optimization problem and can be solved by evolutionary algorithms.

The selection depends on the conditions of the problem and the application. Our circuits follow the same equation for optimization. Then, they are optimized by evolutionary algorithms. In the past, genetic algorithms were applied in order to optimize analogue circuits, but here we try to optimize analogue circuits by genetic algorithms, PSO algorithm and after that a new structure is introduced in order to make use of both of the algorithms simultaneously and in combination with an appropriate fitness function. Nowadays, with the development of electronic circuits due to the need to optimize and speed limits there will be more. Evolutionary algorithms are implemented by means of MATLAB. The fitness function and the limitations are examined by HSPICE, simulation software, and there is a link between the two mentioned software programs for analyzing and optimization. The technology we applied is based on stimulation and it is done as follows: first, the parameters related to the evolutionary algorithms are produced by MATLAB, and they are sent to the circuit simulator HSPICE for optimization and examination. After simulation, these parameters are sent back to MATLAB again so that they are examined by the evolutionary algorithm and then they are optimized. These stages are repeated until we reach the stop point of the algorithm. In Section 2 related work which has been done is listed. Section 3 briefly reviews the genetic algorithm and its flowchart. PSO algorithm and its flow chart are briefly reviewed in Section 4. Section 5 presents stages of the proposed work. In this section, we try to introduce the fitness functions and procedures which have been performed. In Section 6 two circuits and their parameters are listed which will be used in optimization. This section is the most important part because the proposed method is discussed in this section that combines the genetic algorithms and PSO and the simultaneous use of these two methods. Here we analyze the circuits with evolutionary algorithms and then compare them with each other to obtain the best method for designing the analog circuit based on evolutionary algorithms.

2. Related Works

The traditional design of analogue circuits [1, 2] does not have the precision, robustness, and efficacy of today's systems. Therefore, designs based on optimization are applied. In methods that are based on optimization techniques, in comparison with traditional methods, more precision and robustness are achieved. In these methods the problem is stimulated as a function which is minimized in numerical issues, and then it is located in an optimization loop which is probably repeated several times. It is difficult to choose this function, and it must be done with a lot of care.

On the other hand, the function must be written in the closed form, and this may cause the loss of precision and speed. If it could be written in the closed form, this is for one circuit and it is not generalizable.

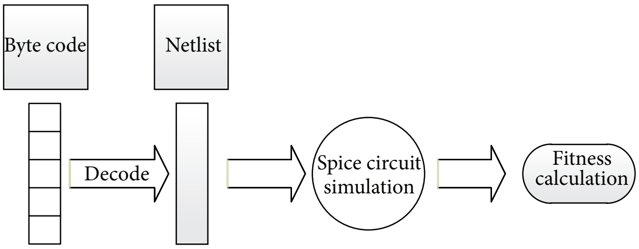


Figure 1: An overview of circuit evaluation process starting with byte coded representation with fitness score.

One of the major limitations in the design of analog circuits is the limited number of parameters that can be investigated. It may be more important when circuit designers want to design circuits with more and more complexity and higher performances and limitations. In many ways a penalty function is used to satisfy these constraints. In these methods, the constrained optimization problem is transformed into an unconstrained one by minimizing the following function [3]:

$$f'_x = f_x + \sum_{i=1}^{n} w_i \left(g_i \left(x\right)\right), \tag{2}$$

where w_i are the penalty coefficients and $g_i(x) > 0$.

The results of these methods are depending on the penalty coefficients and may not be very ideal to the designers. The values of penalty coefficients should have logical values because when they have illogical value, the results will produce infeasible solutions. For finding good solution, the designer usually needs to change penalty coefficients for many times.

Researchers now try to find the ability to handle large-scale and multiobject design problems. Most of the available methods can deal with about 10–20 variables simultaneously, but analog circuits have many unknown variables.

An appropriate location must be selected as the starting point. These stages are shown in Figure 1. However, the process is simpler and precise. In a stage it is attempted to select a good starting point [4] and a better convergence [5].

In recent years, however, evolutionary algorithms have received lots of attention; see, for instance, [6–8] and the references therein. At first, genetic algorithm (GA) was applied, but sometimes this method did not have a favorable outcome. For instance, in GA [9] it was not convergent in the global minimum point.

Ioana Saracut presents an optimization tool for the design of analog circuits with a topology, able to select the best set of passive components values and active devices form within well-defined sets of available values and options. Unlike most of the existing circuit optimization tools, it can search within discrete sets of possible solutions, such as the standard series of values available commercially for passive components and lists of active devices given by the designer [10].

Xuesong Yan says that a major bottleneck in the evolutionary design of electronic circuits is the problem of scale. This refers to the very fast growth of the number of gates, used in the target circuit, as the number of inputs of the evolved logic function increases. This results in a huge search space that is difficult to explore even with evolutionary techniques. Another related obstacle is the time required to calculate the fitness value of a circuit. Use of the traditional genetic algorithm for electronic circuit optimization design is being trapped easily into a local optimum and the convergence speed is slow. They use PSO algorithm to overcome the shortcomings of GA. By analyzing the testing results of four optimization Benchmarks, they reach the following conclusion: in the optimization speed, the PSO algorithm is more efficient than the GA [11].

Fakhfakh et al. [12] proposed particle swarm optimization technique for the optimal design of analog circuits to solve optimization problems [13]. Tlelo-Cuautle et al. [14] presented how we can apply evolutionary algorithms for the synthesis and sizing of analog integrated circuits. Research studies on the examination of analogue circuits with a lot of parameters continue [3].

In order to accommodate a larger design space with a higher number of variables, we may use techniques such as neural networks [15–17]. But here we decide to investigate genetic algorithms and PSO.

3. Genetic Algorithms

These algorithms make use of Darwin's natural selection principles in order to predict or adapt the pattern. Genetic algorithms are often good options for prediction techniques based on randomness. It is briefly stated that genetic algorithm (GA) is a programming technique which uses genetic evolution as a problem-solving pattern. In GA, each unknown parameter is called a gene and the vector of parameters is called chromosome [12]. Initially the chromosomes are generated randomly from the population size. Each chromosome string is made up of integer values to represent a set of parameters to be optimized. The problem to be solved is the input and the solutions are coded according to a pattern which is called fitness function. This function evaluates each selected solution that most of them are selected randomly. First, a primary population (randomly or selectively) is initialized in the scope of problem's limitations and chromosomes are ranked according to their qualifications. The rotating wheel of parents' selection is manufactured according to the qualifications of the chromosomes, and the parents are selected. The new generation generates via mutation and crossover and the stop condition (limited repetition, combination, and convergence examination) and then the examined condition are met, the algorithm is stopped, and the result is announced. If the condition is not met, the stop of chromosomes is ranked again, and the above stages are repeated again. These stages are shown in Figure 2. In Figure 3 we show how we can calculate fitness and its process.

4. PSO Algorithm

The PSO algorithm is a very important optimization algorithm which is inspired by the group movement of birds (and other animals which live in groups). The response of each particle is stated as the location of a particle which has a velocity to approach the location of its food.

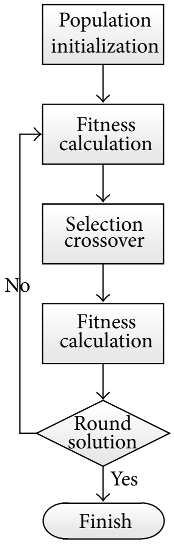


Figure 2: Genetic algorithm flowchart.

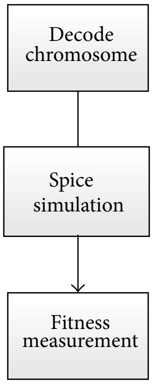


Figure 3: Fitness calculation flowchart.

Each particle generally adjusts its velocity toward its food by means of its own location history, the location history of birds in its specific vicinity, the location history of all the birds in search of the food, and the predetermined inertia. The movement of each particle depends on four factors: (1) the present situation of the particle, (2) the best situation the particle has ever had (*P*best), (3) the best situation which is obtained in the specific vicinity of the particle (*L*best), and (4) the best situation which all the particles have ever had (*G*best); each particle's update follows these relations:

$$\text{Present}\,[\,] = \text{present}\,[\,] + v\,[\,]\,. \tag{3}$$

Here, the last statement specifies the particle's velocity (the amount of location changes for each particle). This statement is calculated in this way:

$$\begin{aligned} v_n = {} & w \times v_n + c_1 \times \text{rand}\,() \times \left(p_{\text{best}_n} - x_n\right) \\ & + c_2 \times \text{rand}\,() \times \left(g_{\text{best}_n} - x_n\right). \end{aligned} \tag{4}$$

Rand () produces the random number in the range of $[0, 1]$; c_1 and c_2 are called "acceleration coefficients." Experience shows that if the number of particles is approximately 10–50, it is enough for convergence in such problems. c_1 is the significance related to the best situation of each particle, and c_2 is the significance related to the best situation of the vicinities. It is usually considered that $c_1 + c_2 = 4$ which is only chosen because of experiential reasons.

The simultaneous application of the evolutionary algorithms can be a way to cover the deficiencies of each algorithm and develop the analysis result. This application can be done in different ways; for instance, a combination of genetic algorithm and PSO is used here. These stages are shown in Figure 4.

5. The Design Optimization of an Op-Amp by means of Evolutionary Algorithms

Circuit parameters are the parameters of the optimization problem and make the chromosomes include the transistors size (length and width) and the amount of compensation capacitor. Four key features of the circuit whose optimized values are more influential in the circuit performance are considered as the fitness functions including DC gain (AV), pw (consuming power), bandwidth, and SR (Slew rate). Our objective is to obtain the appropriate values for circuit parameters maximizing values related to dc gain, bandwidth, GB (Gain Bandwidth), and SR and minimizing consuming power simultaneously.

The implementation of the evolutionary algorithms is done in MATLAB and the calculations of the functions are done by HSPICE. In fact, in this part fitness function is calculated and the inputs applied in evolutionary algorithms are provided. The calculation of the functions for parents and children population which is a step of evolutionary algorithms is done as follows.

First variables are read from MATLAB and they are written in the input file of HSPICE (.sp). Then, HSPICE is run, and finally the values of the fitness function are read from the output file of HSPICE (.lis) and they are sent to MATLAB. The first circuit has 900 input parameters and 4 fitness functions (eight transistors and a capacitor c_l to be adjusted) and the second circuit has 1600 input parameters and 4 fitness functions (fifteen transistors and a capacitor c_l to be adjusted). When only genetic algorithm is applied the population of each swarm is 100 and the number of swarms is 100.

Here we use the three fitness functions and compare the responses obtained and then the best fitness function is introduced. The applied fitness functions are

$$f_i = \sum w_i m_i, \tag{5}$$

$$f_i = \prod m = \frac{(m_1 * m_2 * m_3)}{m_4}, \tag{6}$$

$$f_i = \frac{(m_1^{\alpha_1} * m_2^{\alpha_2} * m_3^{\alpha_3})}{m_4}. \tag{7}$$

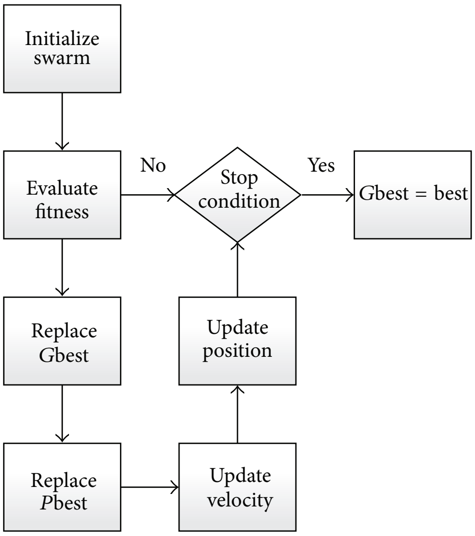


Figure 4: PSO algorithm flowchart.

These functions have been tested in algorithms and the results are listed in the tables for each section separately. In these functions m is the circuit parameters, such as GB and AV, and α is coefficient obtained from the optimization.

6. Proposed Method

Several steps are considered in this section. First, the GA and PSO are selected for optimization. The next step is to design a procedure to link them. HSPICE is selected for simulation of electronic circuits and MATLAB is selected for implementation of evolutionary algorithms. The first step was optimization by GA. At each step, the GA is implemented and will attempt to run HSPICE. HSPICE primary inputs are defined in a GA as the starting point. HSPICE processes the input values and generates the output. HSPICE outputs are considered as MATLAB inputs and MATLAB re-optimized the results. The optimization is determined by the parameters specified in the program code. For example, the starting point of the analysis of an analog circuit is obtained manually. After the first step, a series of analysis again is considered as HSPICE input. These steps also were taken for PSO.

The most important decision part is choosing a new technique to optimize the circuits. Each of these methods has already been used separately, but each has its own advantages and disadvantages. Here we decided to adopt an approach that combines the advantages of the methods. In other words, the disadvantage of this method is covered by benefits on the other methods. For example, the GA has more accuracy and less speed and PSO has more speed and less accuracy in convergence. So the proper use of these two methods can provide the best results.Selected fitness function is a milestone in this project. Fitness functions that have been used are given in (5)–(7). At first it was decided that α_1 to α_n in (7) should be swept. The best reason for using this fitness function is comprehensive equation. But after the reviews, due to the problems such as fluctuation, it was decided to

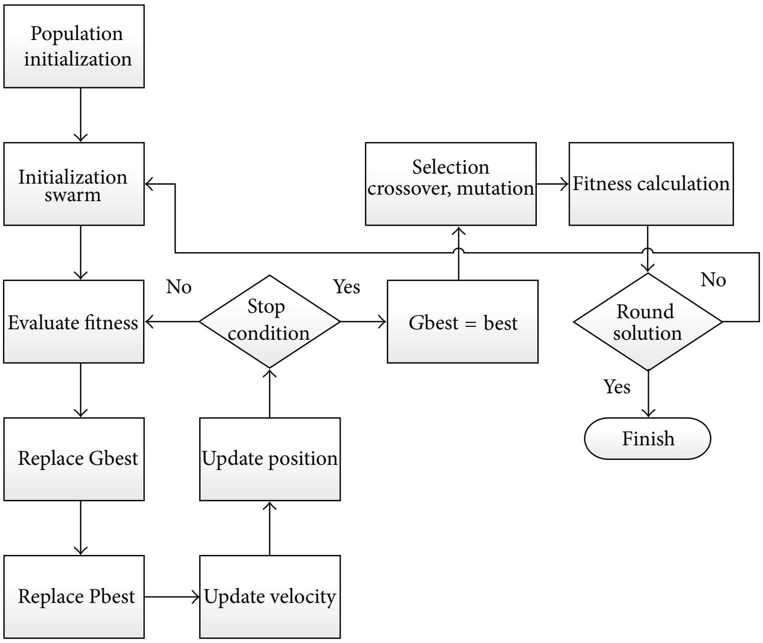


Figure 5: Simultaneous PSO and genetic algorithm flow chart.

use a combination of algorithms. Thus α_1 to α_n with GA and the circuit parameters are optimized by PSO. This procedure is the main point of our work and is shown in Figure 5. After many tests it is found that using this method would obtain much better results compared with when we apply the algorithms individually. Starting points of HSPICE are produced by MATLAB. The output is called by MATLAB and the first step of optimization is completed. In this section, first GA optimizes α parameters and then PSO runs 100 times to optimize the circuit parameters. Then GA is executed and the process is repeated. Analysis of the results obtained from this method compared with the results that were obtained by GA and PSO methods is shown in the tables.

When genetic and PSO algorithms are used simultaneously, the population of each swarm is 100 and the number of the swarms is 50. In this state, algorithm precision is 2^{-8} and mutation probability pm = 0.005 and crossover probability pc = 0.9. Genetic algorithm works in the way that the values of input parameters are proposed by optimization algorithm, and then the circuit is stimulated by those values and the results are derived. After 100 swarms, the response is specified. This is done by 0.6 μm and 0.18 μm technology and the results are compared with each other. All the examples are carried out on a computer with the specifications of 2.5 GHz and 4 G RAM and the running time includes the time consumed by MATLAB and HSPICE and making a link between these two software programs. These stages are shown in Figure 5.

Using various technologies enables us to display the advantages of each method by the software separately and then is considered to be the best choice. If we want to use new technologies in this way it is not needed to make fundamental changes in the program.

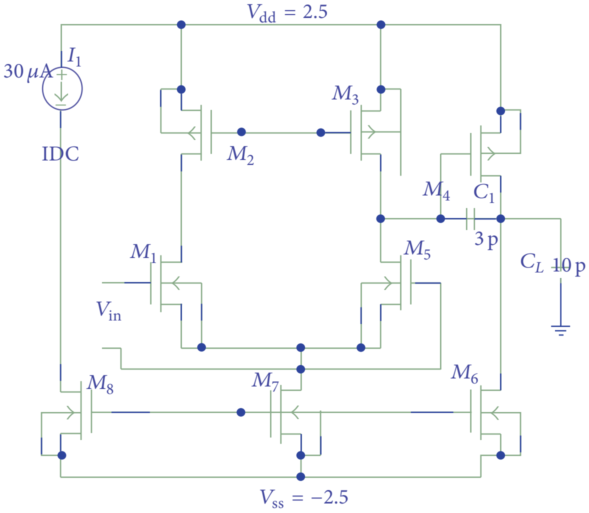


Figure 6: Two-stage amplifier.

6.1. Example 1: Two-Stage Op-Amp Design. A two-stage op-amp is shown in Figure 6 and the selected technologies are 0.6 μm and 0.18 μm. The design parameters are the length of transistors and the amount of compensation capacitor. All these transistors must work in saturation region. In that $V_{DS} > (V_{GS} - V_{TH})$ must be set for NMOS transistors. In every stage, the results of optimization by the algorithms are compared together via different fitness functions.

For both circuits in Figures 6 and 7, the genetic algorithm is implemented with functions equations (5) and (6) of the PSO with the same equations runs.

The main problem after algorithm selection is selecting fitness function. Fitness function mentioned in (7) was firstly

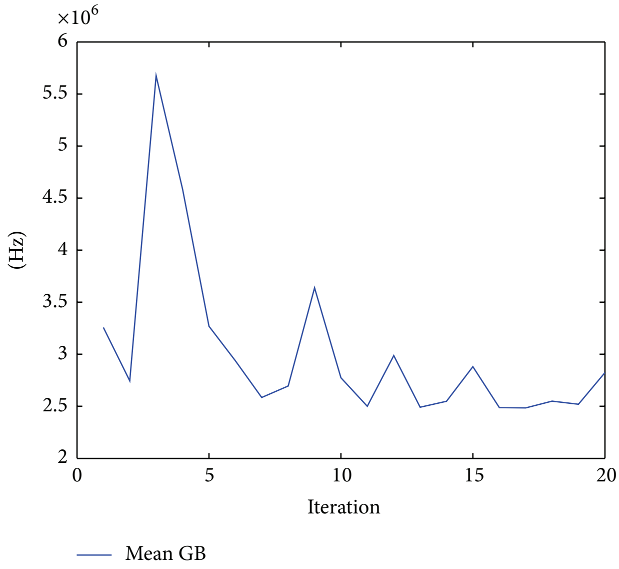


Figure 7: GB value in each stage that PSO algorithm coefficients are swept.

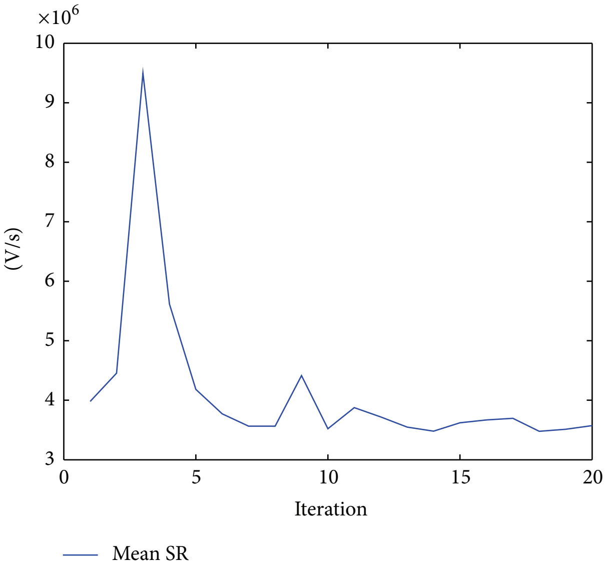


Figure 8: SR value in each stage that PSO algorithm coefficients are swapped.

investigated by PSO algorithm in the way that α parameters are swept at first; in this stage sweep steps with accuracy of 0.1 are considered and responses are recorded in Tables 1 to 4; in the next stage, α parameters are optimized by genetic algorithm, and the circuit parameters are optimized by PSO algorithm. Best results are obtained by selecting the function that is mentioned in (7). This fitness function is actually a series of multiply-and-accumulate operations and it is the complete set. Each of these experiments is undertaken 10 times and the best of them are presented in Tables 1, 2, 3, and 4. Consider

$$\text{FOM} = \frac{\text{AV} * \text{SR} * \text{GB}}{\text{pw}}. \tag{8}$$

In the first stage, if we compare 0.6 μm and 0.18 μm technologies, as we expect, the results obtained from 0.18 μm have far better results. For example, when only genetic algorithm is used, and fitness function is in sum format, FOM responses of 0.18 μm compared to 0.6 μm are approximately 20% better.

When only PSO algorithm is used and fitness function is in sum format, FOM responses of 0.18 μm are 2.25 times of 0.6 μm technology. If fitness function is in multiply format FOM responses obtained from 0.18 μm are 1.699 times of 0.6 μm technology.

When PSO algorithm coefficients are swept, FOM responses obtained from 0.18 μm are 2.82 times of 0.6 μm. The stages of convergence are drawn in Figures 7, 8, and 9.

When genetic and PSO algorithms are used simultaneously FOM responses of 0.18 μm are approximately 1.3 times of 0.6 μm technology. The stages of convergence are drawn in Figures 10, 11, 12, and 13.

In 0.6 μm technology, when PSO and genetic algorithms are used simultaneously, FOM responses are 1.4 times of when only PSO algorithm and fitness function in sum format are used. FOM is also 20 times of when only genetic algorithm is used.

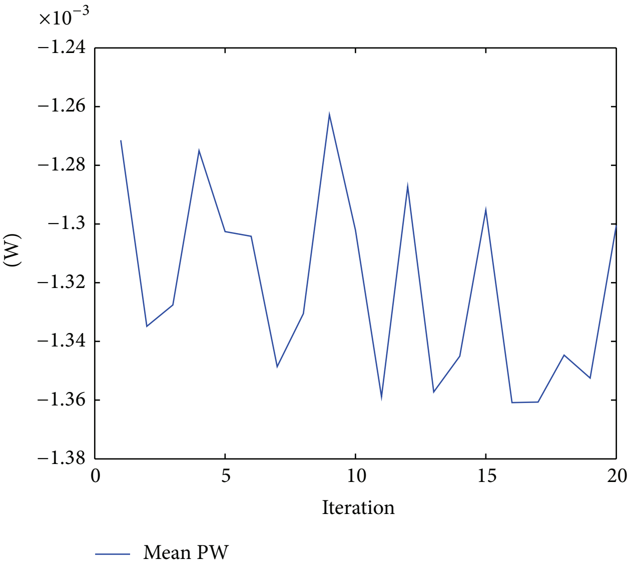


Figure 9: PW value in each stage that PSO algorithm coefficients are swapped.

In 0.18 μm technology, when PSO and genetic algorithms are used simultaneously, FOM responses are 1.008 times of when only PSO algorithm and fitness function in multiply format are used. FOM is also 27.7 times of when only genetic algorithm is used. The application of fitness function in multiply format is better than its application in sum format.

Obviously, when fitness function coefficients are swapped, the response has variation, but when evolutionary algorithms are used in a shared manner, better results are obtained but they are time-consuming.

6.2. Example 2: The Design of Folded Cascade Amplifier. A folded cascade amplifier is shown in Figure 14 and the

Table 1: The results of fitness function stimulation in the sum format and 0.6 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	5*e*5	3.54*e*5	1.832*e*5	6.98*e*4
GB (MHz)	5	8.2	25.7	15.98	67.5
$SR\ (v/\mu s)$	10	1.44	20.7	18.67	56.4
POWER (mw)	2	0.997	1.17	12	2.02
FOM(*e*5)	1.25	59	1609.6	45.5	1315.4

Table 2: The results of fitness function stimulation in the sum format and 0.18 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	7.79*e*5	5.41*e*5	0.88*e*5	0.935*e*4
GB (MHz)	5	6.28	11.8	12.93	62.9
$SR\ (v/\mu s)$	10	1.41	13	18.5	19.4
POWER (mw)	2	1.41	1.16	13	1.15
FOM(*e*5)	1.25	48.9	715.4	16.1	1011.6

Table 3: The results of fitness function stimulation in the multiply format and 0.6 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	10*e*5	3.51*e*5	0.88*e*5	0.935*e*4
GB (MHz)	5	6.3	15.3	12.93	62.9
$SR\ (v/\mu s)$	10	1.2	16.6	18.5	19.4
POWER (mw)	2	1.44	1.162	13	1.15
FOM(*e*5)	1.25	52.5	767.1	16.1	1011.6

Table 4: The results of fitness function stimulation in the multiply format and 0.18 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	4*e*5	4.3*e*5	1.832*e*5	6.98*e*4
GB (MHz)	5	8.18	17.3	15.98	67.5
$SR\ (v/\mu s)$	10	1.45	20.2	18.67	56.4
POWER (mw)	2	0.997	1.165	12	2.02
FOM(*e*5)	1.25	47.5	1304	45.5	1315.4

Table 5: The results of fitness function stimulation in the sum format and 0.6 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	2.159*e*9	2.755*e*7	4*e*9	8.52*e*7
GB (MHz)	10	42.11	14	9.924	18.8
$SR\ (v/\mu s)$	10	10.09	9.732	44.56	10.4
POWER (mw)	50	27.9	60.77	31.6	27.23
FOM(*e*4)	1	32.9*e*5	61*e*2	55.9*e*5	61*e*3

Table 6: The results of fitness function stimulation in the sum format and 0.18 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	2.9*e*9	3.98*e*9	2.74*e*7	4.43*e*8
GB (MHz)	10	41.06	62.74	10.37	99.24
$SR\ (v/\mu s)$	10	10.17	17.06	83.81	44.82
POWER (mw)	50	26.34	13.43	89.26	31.23
FOM(*e*4)	1	46*e*5	317*e*5	26.6*e*3	63*e*5

Table 7: The results of fitness function stimulation in the multiply format and 0.6 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	2.46e9	7.89e9	4e9	8.52e7
GB (MHz)	10	40.88	17.7	9.924	18.8
$SR\,(v/\mu s)$	10	9.73	12.92	44.56	10.4
POWER (mw)	50	25.42	13.66	31.6	27.23
FOM(e4)	1	38.4e5	132e5	56e5	61e3

Table 8: The results of fitness function stimulation in the multiply format and 0.18 μm technology.

Specification	Constraint	GA	PSO	PSO(α)	GA + PSO
DC gain	5000	1.9e9	3.7e9	2.74e7	4.429e8
GB (MHz)	10	41.0	60.24	10.37	99.24
$SR\,(v/\mu s)$	10	10.1	18.1	83.81	44.82
POWER (mw)	50	26.3	25.6	89.26	31.23
FOM(e4)	1	30e5	158e5	26.6e3	63e5

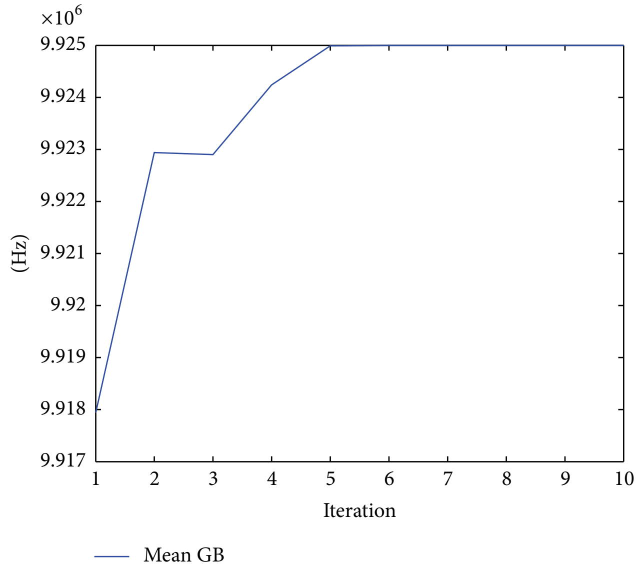


Figure 10: GB value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

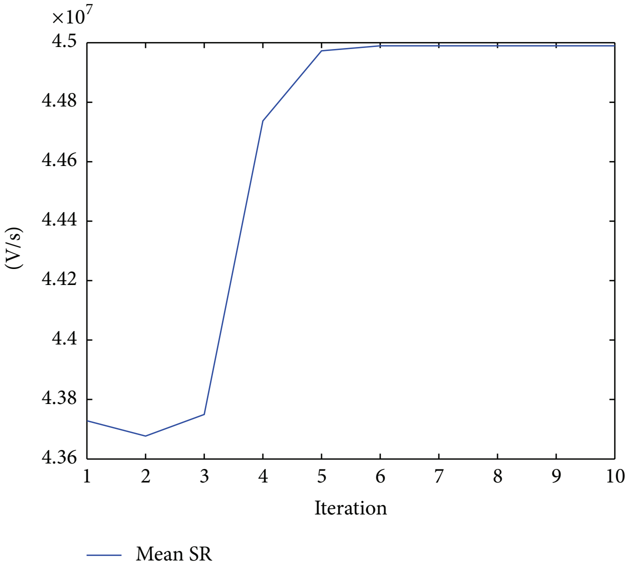


Figure 11: SR value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

applied technologies are 0.6 μm and 0.18 μm. The design parameters are the length of transistors and the amount of compensation capacitor. All these transistors must work in saturation region. That is, $V_{DS} > (V_{GS} - V_{TH})$ must be set for NMOS transistors. In each stage, the results of optimization of algorithms are compared by means of different fitness functions. Fitness functions are the same functions used in the previous example.

Here, also fitness function (5) was firstly investigated by PSO algorithm in the ways that α parameters are swept at first, and in the next stage, α parameters are optimized by genetic algorithm, and circuit parameters are optimized by PSO algorithm. Each of these experiments is undertaken 10 times and the best of them are presented in Tables 5, 6, 7, and 8. The stages of convergence are drawn in Figures 15, 16, 17, 18, 19, and 20.

In the first stage, if we compare 0.6 μm and 0.18 μm technologies, as we expect, the results obtained from 0.18 μm have far better specifications. For example, when only genetic algorithm is used, and fitness function is in sum format, FOM responses of 0.18 μm compared to 0.6 μm are approximately 41% better.

When only PSO algorithm is used and fitness function is in sum format, FOM responses of 0.18 μm are 5000 times of 0.6 μm technology. If fitness function is in multiply format FOM responses obtained from 0.18 μm are 1.199 times of 0.6 μm technology. When PSO and genetic algorithms are used simultaneously, FOM responses of 0.18 μm are 1.032 times of 0.6 μm technology.

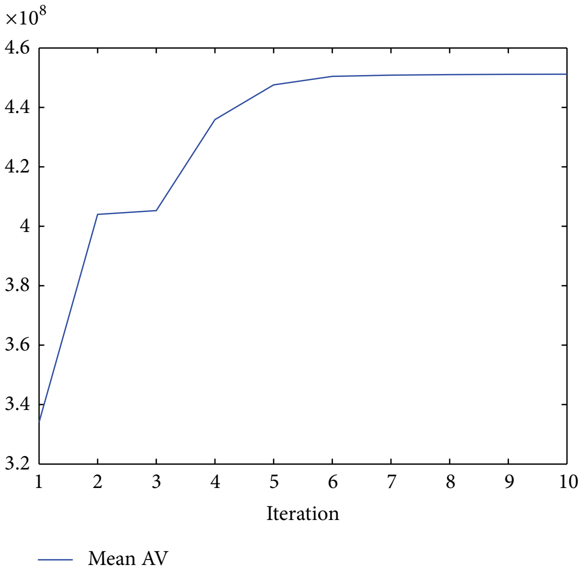


Figure 12: Gain value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

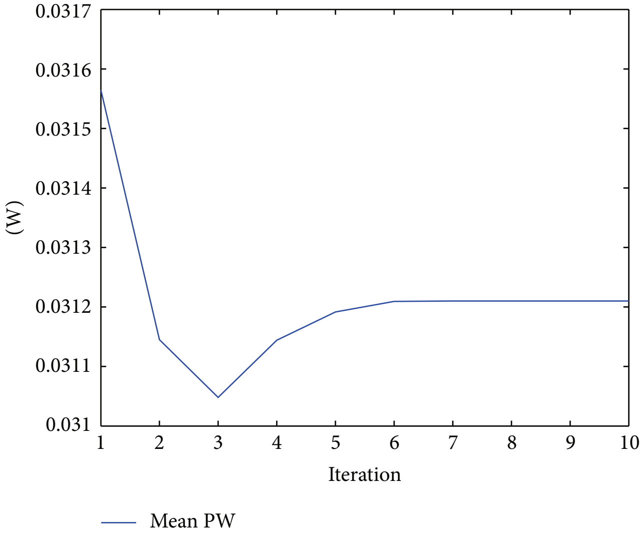


Figure 13: PW value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

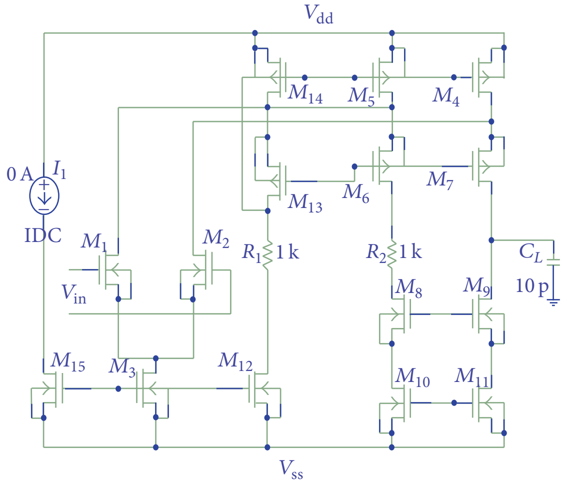


Figure 14: Folded cascade op-amp.

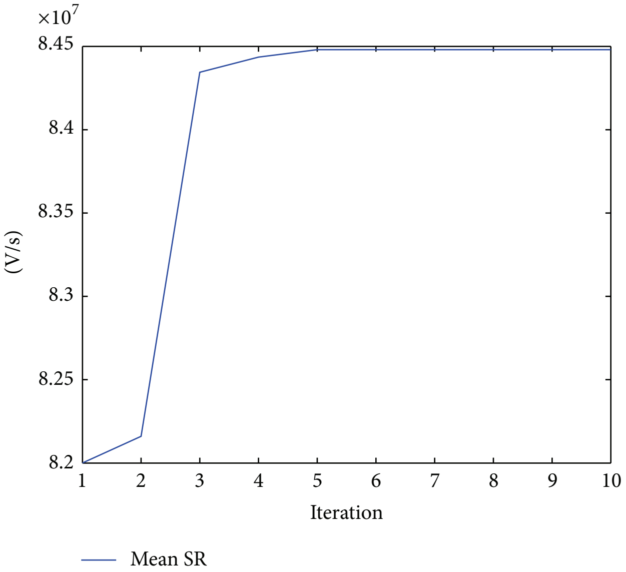


Figure 15: SR value in each stage that PSO algorithm coefficients are swapped.

In 0.6 μm technology, when PSO and genetic algorithms are used simultaneously, FOM responses are 10 times of when only PSO algorithm and fitness function in sum format are used. In 0.18 μm technology, when PSO and genetic algorithms are used simultaneously, FOM responses are 2.1 times of when only genetic algorithm and fitness function in multiply format are used. The application of fitness function in multiply format is better than its application in sum format.

As it is clear from the figures, when PSO and genetic algorithms are applied simultaneously, the results are approximately convergent after 7 stages which indicate the appropriate selection of fitness function, but when αs are swapped, there may be variation and it is not a good result. Checking the results indicates that using a combination of optimization algorithms with selecting the appropriate fitness function in the optimization process highly required being effective. Each of these algorithms has a prominent feature, for example, different speed or accuracy that better results can be achieved by combining them.

Using a combination of genetic algorithms and PSO simultaneously gives us the best results compared with using these algorithms separately.

7. Conclusion

This method is a new look to design and optimize analog electronic circuits that enables users by taking advantage of

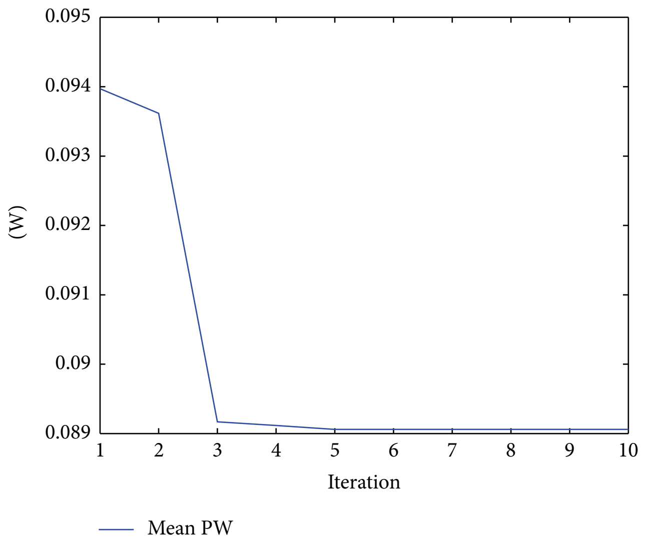


FIGURE 16: PW value in each stage that PSO algorithm coefficients are swapped.

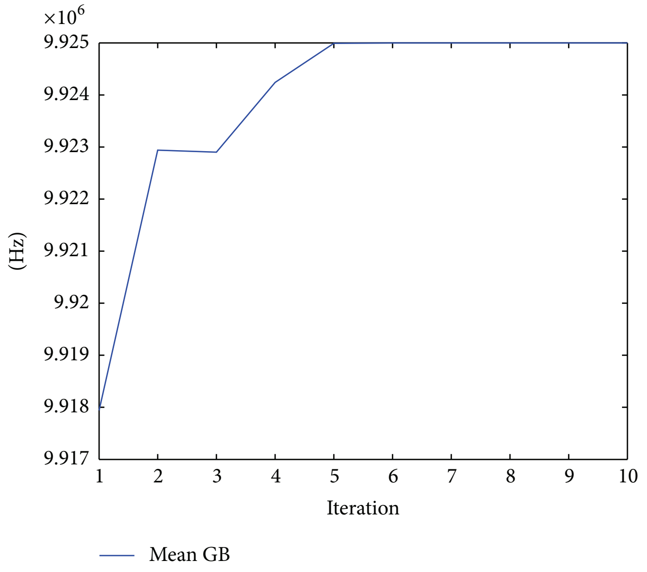


FIGURE 17: GB value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

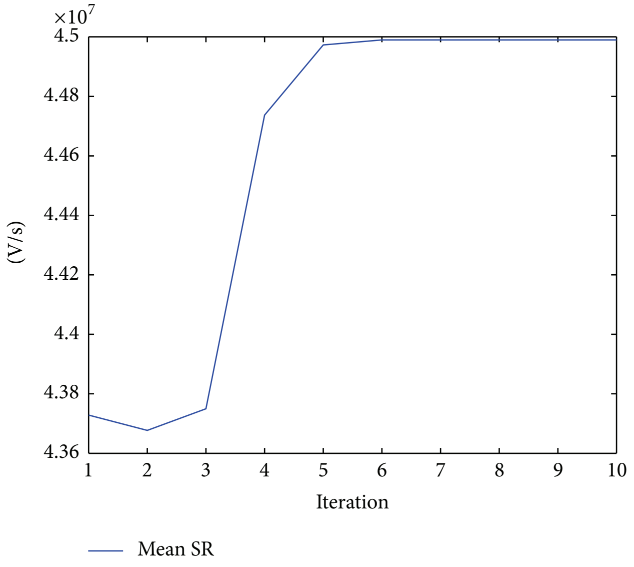


FIGURE 18: SR value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

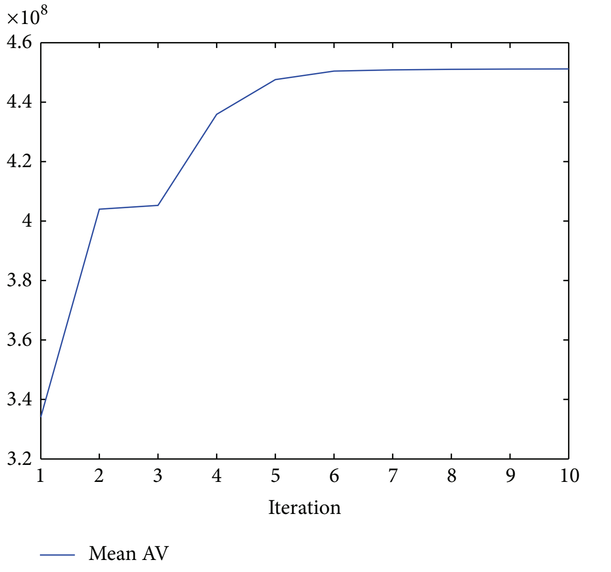


FIGURE 19: AV value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

software to design hardware and reduce the design complexity.

The circuit design has been performed by SPICE simulator. This paper investigated two methods of evolutionary algorithms, namely, genetic algorithm (GA) and PSO algorithm and their combination in order to optimize analogue circuits.

It was shown that combination of both algorithms has better results. Coefficients are based on GA and parameters are optimized by PSO algorithm. The reason for this choice is that the PSO algorithm can converge faster than GA because the number of its operators is less than that of GA. The application of a combination of these two algorithms has the following advantages.

(1) More precision in search space and not being trapped in the local minimum.

(2) Satisfaction of the designer's desire specifications and its limitations.

(3) Being appropriate for circuits with lots of elements.

(4) Using this method can meet the designer's specifications for highly constrained problems.

(5) It can be suitable for large-scale problems.

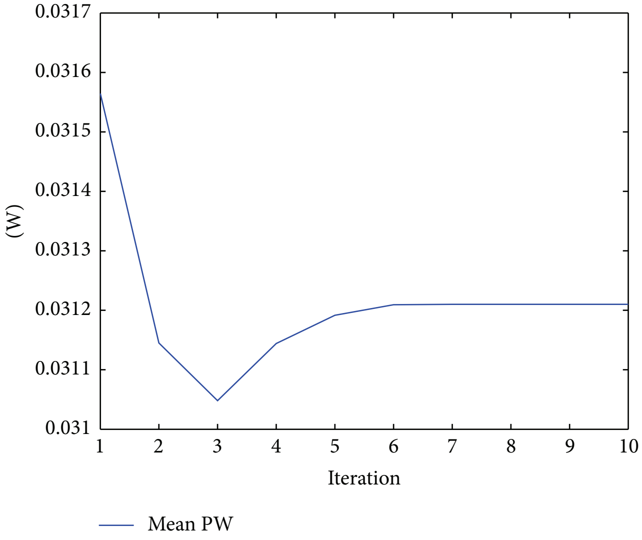


Figure 20: PW value in one stage, when PSO algorithm coefficients are selected by genetic algorithm.

Evolutionary algorithm includes a wide class of algorithms inspired from nature and function (such as harmonic search algorithm, ant colony, worms, greedy, and gravity). More diverse set of algorithms can be used for further research in order to select the appropriate method for the design of analog circuits is proposed. Other interesting aspects of this work are (1) the use of real time methods and considering the delay effects [18–20]; (2) the use of any electronic CAD software which improves the process of design and fabrication. In future also we can use other cost functions and software such as cadence.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from the Polish-Norwegian Research Programme operated by the National Centre for Research and 24 Development under the Norwegian Financial Mechanism 2009–2014 in the frame of Project Contract no. Pol-Nor/200957/47/2013.

References

[1] M. G. R. Degrauwe, O. Nys, E. Dijkstra et al., "IDAC: an interactive design tool for analog CMOS circuits," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 6, pp. 1106–1116, 1987.

[2] C. A. Makris and C. Toumazou, "Analog IC design automation. Part II. Automated circuit correction by qualitative reasoning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 2, pp. 239–254, 1995.

[3] B. Liu, Y. Wang, Z. Yu et al., "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration, The VLSI Journal*, vol. 42, pp. 137–148, 2009.

[4] G. Stehr, M. Pronath, F. Schenkel, H. Graeb, and K. Antreich, "Initial sizing of analog integrated circuits by centering within topology-given implicit specifications," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD '03)*, pp. 241–246, November 2003.

[5] M. del Mar Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a CMOS op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, pp. 1–21, 2001.

[6] D. Nam, Y. D. Seo, L. J. Park, C. H. Park, and B. Kim, "Parameter optimization of an on-chip voltage reference circuit using evolutionary programming," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 4, pp. 414–421, 2001.

[7] S. Balkir, G. Dündar, and G. Alpaydm, "Evolution based synthesis of analog integrated circuits and systems," in *Proceedings of the NASA/DOD Conference on Evolvable Hardware*, pp. 26–29, June 2004.

[8] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS opamp synthesis by means of a genetic algorithm," in *Proceedings of the 32nd Design Automation Conference*, pp. 433–438, June 1995.

[9] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[10] I. Saracut, M. Neag, R. Onet, and I. Kovacs, "Design optimizationof analog active circuits using the genetic algorithm," *Actatechnicanapocensis Electronics and Telecommunications*, vol. 53, no. 4, 2012.

[11] Y. A. N. Xuesong, H. U. Chengyu, Y. A. O. Hong, F. A. N. Yuanyuan, Qingzhong LIANG, and L. I. U. Chao, *Electronic Circuit Optimization Design Algorithm Based on Particle Swarm Optimization*, PRZEGLAD ELEKTROTECHNICZNY, 2013.

[12] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog Integrated Circuits and Signal Processing*, vol. 63, no. 1, pp. 71–82, 2010.

[13] P. Prem Kumar, K. Duraiswamy, and A. Jose Anand, "An optimized device sizing of analog circuits using genetic algorithm," *European Journal of Scientific Research*, vol. 69, no. 3, pp. 441–448, 2012.

[14] E. Tlelo-Cuautle, I. Guerra-Gómez, M. A. Duarte-Villaseñor et al., "Applications of evolutionary algorithms in the design automation of analog integrated circuits," *Journal of Applied Sciences*, vol. 10, no. 17, pp. 1859–1872, 2010.

[15] O. Garitselov, S. P. Mohanty, and E. Kougianos, "Fast-accurate non-polynomial metamodeling for nano-CMOS PLL design optimization," in *Proceedings of the 25th International Conference on VLSI Design (VLSID '12)*, pp. 316–321, January 2012.

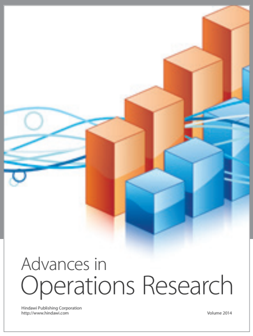
[16] O. Okobiah, S. P. Mohanty, E. Kougianos, and O. Garitselov, "Kriging-assisted ultra-fast simulated-annealing optimization of a clamped bitline sense amplifier," in *Proceedings of the 25th International Conference on VLSI Design (VLSID '12)*, pp. 310–315, January 2012.

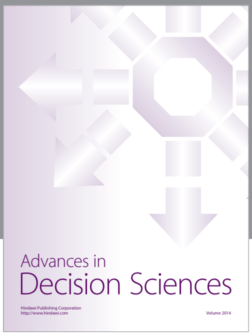
[17] H. R. Karimi and H. Gao, "New delay-dependent exponential H∞ synchronization for uncertain neural networks with mixed time delays," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 40, no. 1, pp. 173–185, 2010.

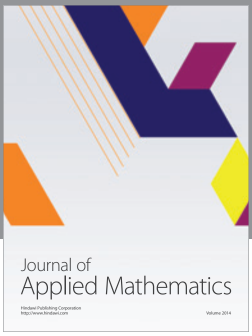
[18] H. R. Karimi, "Robust delay-dependent H_∞ control of uncertain time-delay systems with mixed neutral, discrete, and distributed time-delays and Markovian switching parameters," *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 58, no. 8, pp. 1910–1923, 2011.

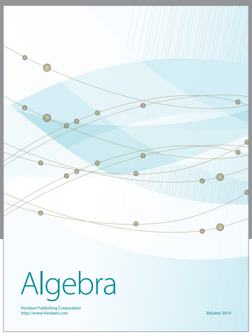
[19] S. Huang, Z. Xiang, and H. R. Karimi, "Robust l_2-gain control for 2D nonlinear stochastic systems with time-varying delays and actuator saturation," *Journal of the Franklin Institute. Engineering and Applied Mathematics*, vol. 350, no. 7, pp. 1865–1885, 2013.

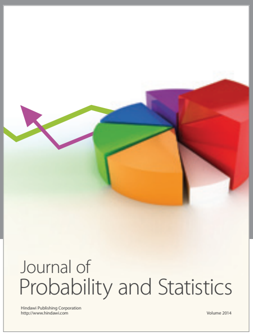
[20] Z. Li, H. Gao, and H. R. Karimi, "Stability analysis and H_∞ controller synthesis of discrete-time switched systems with time delay," *Systems & Control Letters*, vol. 66, pp. 85–93, 2014.











Hindawi

Submit your manuscripts at
http://www.hindawi.com