Journal of Theoretical and Applied Information Technology
20th June 2014. Vol. 64 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GA AND ACO TECHNIQUES FOR THE ANALOG CIRCUITS DESIGN OPTIMIZATION

[1] BACHIR BENHALA, [2] OMAR BOUATTANE
[1] Sensor Electronic & Instrumentation Group, LEAB, Faculty of Sciences,
University of Moulay Ismaïl, B.P. 11201, Zitoune, Meknes, Morocco.
[2] Laboratory of Signals, Distributed Systems and Artificial Intelligence, LSSDIA
University Hassan II Mohammedia Casablanca, ENSET, B.P.159, Mohammedia, Morocco.
E-mail: [1] benhalab@yahoo.fr , [2] o.bouattane@gmail.com

ABSTRACT

In this paper we propose a comparison between two population-based metaheuristic techniques, the Genetic Algorithm (GA) and the Ant Colony Optimization (ACO), to make easier the sizing of analog circuits with optimal objective functions. The paper details the corresponding algorithms and highlights the optimal design of a positive second generation current conveyor (CCII+) and an active filter circuit. The computing time and robustness of both algorithms are checked. SPICE simulation is used to validate the obtained sizing/performances.

Keywords: *Metaheuristic, Ant Colony Optimization, Genetic Algorithm, Current Conveyors, Second Order Low-pass Filter.*

1. INTRODUCTION

Over the past decade, important progress in the combinatorial problem resolution has been achieved with the appearance of a new generation of powerful and approximate optimization methods, known as metaheuristics [1]. These methods lead to solve real-world problems within an acceptable length of time. They always offer "good" approximation of the "best" solutions for optimization problems [2]. The microelectronic field development was also investigated by those methods. Thus, some (meta)heuristics were used by the designers to optimize the sizing of the analog components automatically, such as Tabu Search (TS) [3], Genetic Algorithms (GA) [4], Local Search (LS) [5], Wasp Nets (WN) [6], Particle Swarm Optimization (PSO) [7] and recently Ant Colony Optimization (ACO) [8,9].

The discrete components are still preferred in analog active filter design. In order to reduce the costs and make the design more reliable, discrete components such as resistors and capacitors are chosen from the industrial series values such as E12, E24, E48 series. Performing an exhaustive search on all possible combinations of preferred values for obtaining an optimized design is not feasible. Therefore, intelligent search methods must be developed that requires short computation time with high accuracy.

In this work, we focus on the use of the two algorithms: Genetic Algorithm and and Ant Colony Optimization; to solve typical analog circuit sizing problems. Two application examples are considered, a positive second generation current conveyor and a second order low-pass filter. The aim is to compare these two techniques in terms of results quality, robustness and computing time. The SPICE simulations are given to show the validity of obtained results.

The remainder of the paper is structured as follows: The second section presents an overview of the used algorithms. The third section illustrates the two application examples and the fourth section, deals with the simulations and comparison results. The final section is devoted to some concluding remarks.

2. ACO AND GA: A BREF OVERVIEW

2.1 Ant Colony Optimization technique

The ACO technique is inspired by the collective behavior of deposition and monitoring of some traces as it is observed in insect colonies [10], such as ants. It is for example well known that ants deposit pheromone on the ground in order to mark some favorable paths that should be followed by

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

other members of the colony. The ants behavior is used by the scientists to model some optimization problems and sort out some optimal solutions. As an example, for the minimal path finding problem, the modeling approach is established as follows:

- For solving such problems, ants randomly select the vertex to be visited. When an ant k is in the vertex i, the probability for going to the vertex j is given by the following expression [11, 12]:

$$p_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}}{\sum_{l \in J_i^k} (\tau_{il})^{\alpha}.(\eta_{il})^{\beta}} & if \quad j \in J_i^k \\ 0 & if \quad j \notin J_i^k \end{cases} \qquad (1)$$

where J_i^k is the set of neighbours of the vertex i of the k^{th} ant, τ_{ij} is the amount of pheromone trail on the edge (i, j), α and β are weightings parameters that control the pheromone trail τ_{ij} and the visibility value, η_{ij} given by:

$$\eta_{ij} = \frac{1}{d_{ij}} \qquad (2)$$

where d_{ij} is the distance between vertices i and j.

- The pheromone rate values are updated during each iteration by all the m ants that have built a solution in the iteration itself. The pheromone rate τ_{ij}, which is associated with the edge joining vertices i and j, is updated as follows:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (3)$$

Where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}(t)$ is the quantity of pheromone laid 'deposited, or dropped of' on edge (i, j) by ant k:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L^k} & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour,} \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Q is a constant and L_k is the length of the tour constructed by the ant k.

The pseudo code of the ACO procedure is as follows:

```
Random initialization of the pheromone value
Do
 For each iteration
  For each ant
      Compute of the probability P according to (1)
      Determine the Pmax
  End
      Compute OF
 End
     Deduce the best OF
     Update pheromone values according to (3)
End
  Report the best solution
END
```

ALGORITHM 2. PSEUDO CODE OF ACO

2.2 Genetic Algorithm

The GA find their origins in the biological processes of survival and adaptation. Its principle consists of sampling a population of potential solutions. A population of individuals is, initially, randomly generated. The GA performs then operations of selection, crossover and mutation on the individuals, corresponding respectively to the principal of survival of the fittest, recombination of genetic material and random mutation observed in nature [13]. The optimization process is carried out through the generation of successive populations until a stop criterion is met.

To implement the genetic algorithm technique, the following parameters need to be selected are [14]:

- Population size,
- Probability of crossover,
- Probability of mutation.

The pseudo code of the GA procedure is as follows:

```
Random initialization of the population
     max_fitness := 0
Do
For each member chromosome
fitness := Fitness_Evaluation (chomosome)
      If fitness > max_fitness
max_fitness := fitness
fittest_solution = chromosome
      End if
     End for
     While generation < max_generations
offspring := Selection (parents)
fitness := Fitness_Evaluation (offspring)
       If fitness > max_fitness
       max_fitness := fitness
       fittest_solution = offspring
       End if
savefittest_solution
END
```

ALGORITHM 1. PSEUDO CODE OF GA

Journal of Theoretical and Applied Information Technology
20th June 2014. Vol. 64 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. APPLICATION TO THE OPTIMAL DESIGN OF ANALOG CIRCUITS

In this section we present the two circuit's optimization subject: The CCII+ and a second order low-pass filter.

3.1 Performance sizing of a CMOS CCII+

The CMOS positive second generation current conveyor (CCII+) circuit is shown in the following figure:

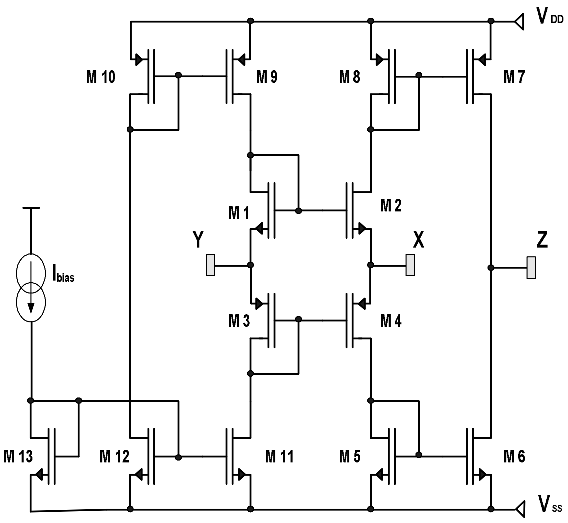

Figure 1: A positive second generation current conveyor (CCII+)

The objective functions to be optimized are:

- R_X: the X-port input parasitic resistance to be minimized.

$$Rx = \frac{1}{gm_2 + gm_4} \quad (5)$$

- *fci*: the current high cut off frequency to be maximized.

$$f_{ci} = \frac{1}{2\pi}\sqrt{\frac{gm_5(gm_4 + go_4)}{Cgs_4(Cgs_5 + Cgs_6 + Cgd_4)}} \quad (6)$$

Cgs, *Cgd*, *gm* and *go* refer to the parasitic grid to source capacitance, the parasitic grid to drain capacitance, the transconductance and the conductance of the MOS transistor, respectively.

The constraints of the problem, correspond to the conditions of saturation of transistors, are presented by expressions (7) and (8):

$$\frac{V_{DD}}{2} + V_{TP} + \sqrt{\frac{2I_{bias}}{\mu_N C_{ox} W_2 / L_2}} \geq \sqrt{\frac{2I_{bias}}{\mu_p C_{ox} W_8 / L_8}} \quad (7)$$

$$\frac{V_{DD}}{2} - V_{TN} - \sqrt{\frac{2I_{bias}}{\mu_N C_{ox} W_5 / L_5}} \geq \sqrt{\frac{2I_{bias}}{\mu_p C_{ox} W_4 / L_4}} \quad (8)$$

where, C_{OX}, V_{TP}, V_{TN}, μ_N and μ_P are technological parameters.

The Bias current I_{bias}=100 µA and the Supply voltages used are V_{SS}/V_{DD} = -2.5V/2.5V.

The geometric variables that will be used to optimize performances of a CMOS positive second generation current conveyor (CCII+) are the MOS transistors sizes. Precisely, they are the channels lengths (LN, and LP) and gates widths (WN and WP) while respecting the saturation conditions of the transistors MOS.

3.1 Performance Optimization Of A Second Order Low-Pass Filter

The considered circuit is a low pass filter formed by the second-order six resistors and two capacitors. The schematic of this filter is given in Figure 2.

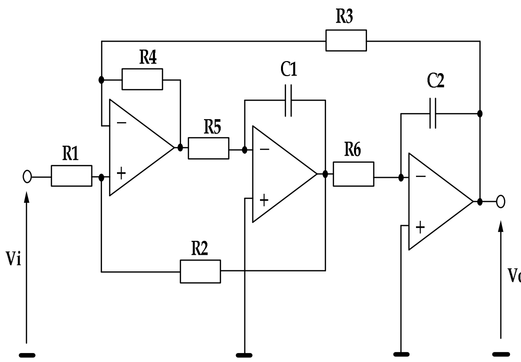

Figure 2: Second order low-pass filter

The cutoff frequency ω and the selectivity factor Q of filter, which depend only on the values of the passives components, are given as follows:

$$\omega = \sqrt{\frac{R_4}{R_3}\left(\frac{1}{C_1 C_2 R_5 R_6}\right)} \quad (9)$$

$$Q = \frac{R_3(R_1 + R_2)}{R_1(R_3 + R_4)}\sqrt{\frac{C_1 R_4 R_5}{C_2 R_3 R_6}} \quad (10)$$

Journal of Theoretical and Applied Information Technology
20th June 2014. Vol. 64 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The specification chosen here is ω_o = 10 000 rad/s (f = 1591.55 Hz) and Q = 0.707 for reduced peak on low-pass response.

The values of the resistors and capacitors to choose must be able to generate ω and Q approaching the specified values. For this, we define the Total error which expresses the offset values, of the cut-off frequency and the selectivity factor, compared to the desired values, by:

$$Total\_error = 0.5\Delta\omega + 0.5\Delta Q \qquad (11)$$

were:

$$\Delta\omega = \frac{|\omega - \omega_o|}{\omega_o}, \qquad \Delta Q = \frac{|Q - 0.707|}{0.707} \qquad (12)$$

The objective function considered is the total error. The decision variables are the resistors and capacitors forming the circuit. Each component must have a value of the standard series (E12, E24, E48 or E96). The resistors have values in the range of 10^3 to $10^6\Omega$. Similarly, each capacitor must have a value in the range of 10^{-9} to 10^{-6}F.

4. SIMULATIONS AND COMPARISON RESULTS

In this section we applied the two algorithms (GA and ACO) to perform optimization of the CCII+ and a second order low-pass filter.

The studied algorithms parameters are given in Table 1 with a generation algorithm of 1000. The optimization techniques work on MATLAB codes and are able to link SPICE to measure performances.

Table 1: The algorithm parameters

ACO	Number of Ants	100
	Evaporation rate (ρ)	0.1
	Quantity of deposit pheromone (Q)	0.2
	Pheromone Factor (α)	1
	Heuristics Factor (β)	1
GA	Population size	100
	Crossover Probability	0.9
	Mutation Probability	0.0001

The simulations are performed using the technology of 0.35 µm CMOS from AMS.

4.1 The Current Conveyor

Tables 2 and 3 give optimal results obtained by using the ACO and GA algorithms for the parameters and the circuit's performances for CCII+.

Table 2: Optimization and simulation results for Rxmin

	LN (µm)	WN (µm)	LP (µm)	WP (µm)	Rx min (Ω)	
					Opt.	Sim.
ACO	0.55	20.76	0.37	30.00	443	464
GA	0.57	15.83	0.39	28.51	512	548

Table 3: Optimization and simulation results for fci_{max}

	LN (µm)	WN (µm)	LP (µm)	WP (µm)	fci_{max} (GHz)	
					Opt.	Sim.
ACO	0.55	05.10	0.35	08.91	1.792	1.787
GA	0.60	05.98	0.36	10.67	1.541	1.546

Figures 3 and 4 show the SPICE simulation results (*Rx* and *fci*) using the obtained optimal values by the ACO and GA algorithms for CCII+. We notice that simulation results are in good agreement with those obtained using the two algorithms.

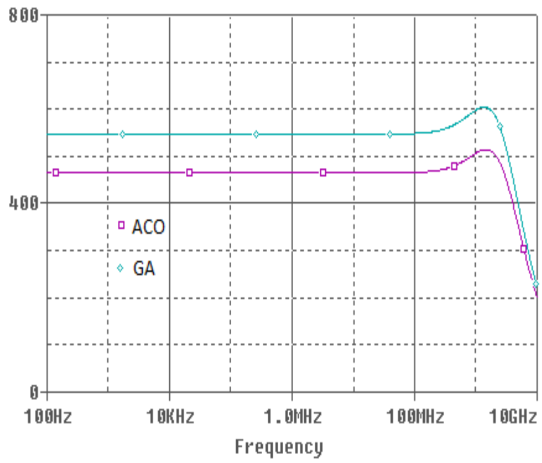

Figure 3: Rx-pole resistance (ohm) vs. frequency (Hz)

Journal of Theoretical and Applied Information Technology
20th June 2014. Vol. 64 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

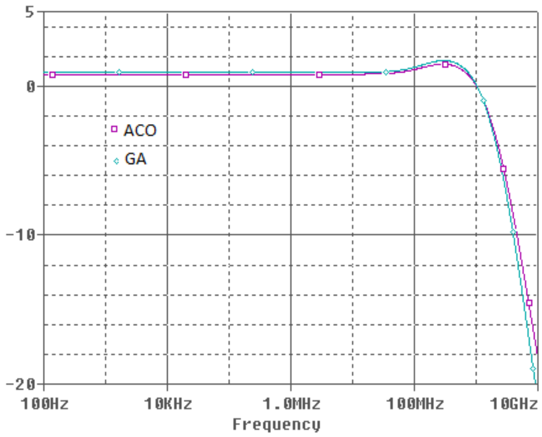

Figure 4: Current gain (dB) vs. frequency (Hz)

4.2 The Second Order Low-Pass Filter

The optimal values of resistors and capacitors forming the considered filter and the performance associated with these values for the different series are shown in Tables 4 and 5 respectively for the GA and the ACO algorithms.

Table 4: Values of components and related filter performance for GA

	E12	E24	E48	E96
R1 (KΩ)	47.0	51.0	53.6	54.9
R2 (KΩ)	100	91.0	95.3	97.6
R3 (KΩ)	47.0	51.0	51.1	51.1
R4 (KΩ)	100	91.0	95.3	93.1
R5 (KΩ)	10.0	9.10	9.53	9.76
R6 (KΩ)	47.0	47.0	48.7	49.9
C1 (nF)	8.20	7.50	7.87	7.68
C2 (nF)	4.70	4.70	4.87	4.99
Δω	0.0773	0.0808	0.0234	0.0121
ΔQ	0.2045	0.0478	0.0506	0.0302
Total error	0.1409	0.0643	0.0370	0.0212

Table 5: Values of components and related filter performance for ACO

	E12	E24	E48	E96
R1 (KΩ)	10.0	9.10	8.66	8.66
R2 (KΩ)	2.20	2.20	2.26	2.26
R3 (KΩ)	5.60	6.20	5.90	6.04
R4 (KΩ)	3.90	3.90	4.22	4.22
R5 (KΩ)	6.80	6.80	6.81	6.98
R6 (KΩ)	1.20	1.30	1.27	1.24
C1 (nF)	15.0	13.0	14.0	13.7
C2 (nF)	56.0	56.0	59.0	59.0
Δω	0.0079	0.0115	0.0006	0.0007
ΔQ	0.0438	0.0612	0.0081	0.0034
Total error	0.0259	0.0364	0.0044	0.0020

Notice that for both algorithms, the accuracy of the spacing values associated to the components affect significantly the performance of the filter. Indeed, the values of the E96 series are the smallest total error compared to other series. For all series, the ACO returned values of components that have a total error smaller than that given by the GA.

In order to check the validity of the results, the following figure shows the PSPICE simulation in the filter gain for the optimal values of the E96 series for the ACO and GA. The cut off frequency are equal to 1593 Hz for the ACO and 1610 Hz for the GA.

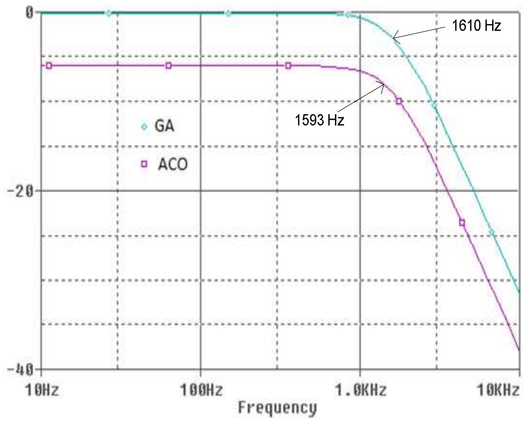

Figure 5: Frequency responses of second order low-pass filter

The following table shows the comparison between the theoretical values and those practices for the error on the cut-off frequency.

Table 6: Comparisons

	GA		ACO	
	Opt.	Sim.	Opt.	Sim.
Δω	0.0121	0.0116	0.0007	0.0009

From the results presented in Table 6, we notice that simulation results are in good agreement with those obtained using ACO and GA.

4.3 Computing Time And Robustness

To complete the comparison, we check the running time and the convergence rate of the two algorithms. The convergence rate is a robustness test which shows the ability of the algorithm to find the same result for different executions.

Table 7 correspond to a comparison between computing times (average running times for 100

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

runs of each algorithm) of the optimization algorithms using a Pentium R - dual Core CPU T4500 - 2. 3GHZ - 2Mo RAM PC.

Table 7: Comparison of the computing time (seconds)

	GA	ACO
Rx min	23.47	112.21
fci max	29.17	123.48
Total error (for E96)	15.72	049.73

We notice that the GA algorithm is faster than the ACO algorithm.

In order to check the convergence rate, the algorithms are repeated a hundred times for optimizing the *Rx*, *fci* and the *Total error (for E96)* objectives. In the Figures 6, 7 and 8 we present the obtained results (respectively for *Rx, fci* and *Total error*) for the ACO and GA algorithms.

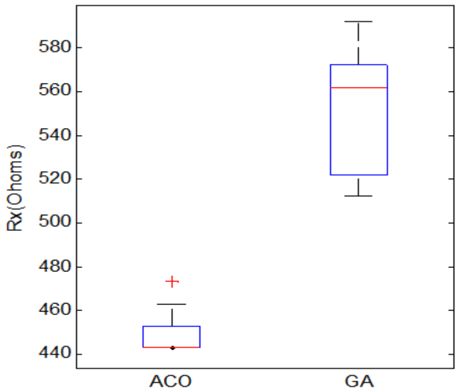

Figure 6: Results obtained for 100 generations for Rx

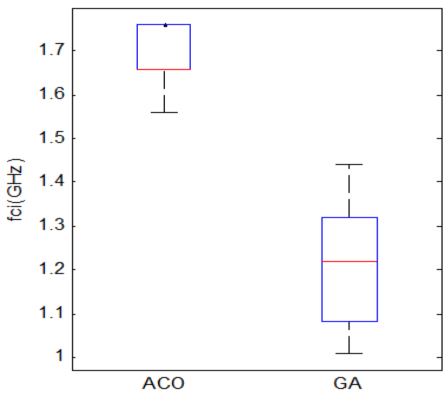

Figure 7: Results obtained for 100 generations for fci

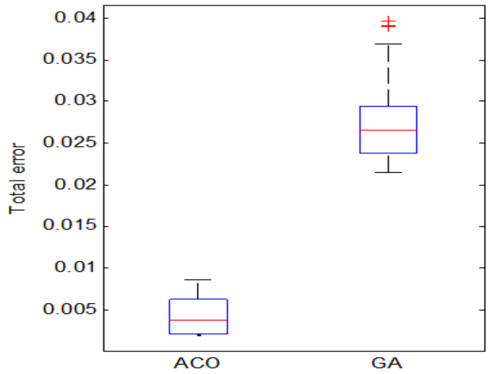

Figure 8: Results obtained for 100 generations for Total error (for E96)

The good convergence ratio can be easily noticed, despite the probabilistic aspect of the two algorithms. We can, also, notice that the robustness of the ACO algorithm is better than the robustness of the GA algorithm; in fact the convergence rates to the same optimal value are 47% and 12% respectively for ACO and GA.

For an overall comparison, the following table resumes and compares the main features of the ACO and GA algorithms.

Table 8: Performances comparison between ACO and GA algorithms

Algorithms	Running time	Robustness	Optimum
ACO	-	+	+
GA	+	- +	- +

'+': good, '- +': medium and '-':low.

5. CONCLUSION

We presented in this paper two metaheuristic optimization techniques that are the Ant Colony optimization and Genetic Algorithm for optimal analog circuits design. Both techniques were used for the optimal sizing of two analog circuits; a CMOS second generation current conveyor and a second order low-pass filter. It has been shown that ACO algorithm offers better results in terms of optimality and robustness. The GA is faster and requires less algorithm-parameters to handle. Accordingly, the choice between these algorithms will depend on the desiderata of the designer. However, our results suggest that a hybrid algorithm consisting of at least two techniques, one taking care of optimum quality, the other taking care of running time, is a promising research direction.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] C.R. Reeves, “Modern heuristic techniques for combinatorial problems”, *Blackwell Scientific Publications,* Oxford, 1993.

[2] I.H. Osman, and J.P. Kelly, “Meta-heuristics: theory and applications”, *Kluwers Academic Publishers,* Boston, 1996.

[3] F. Glover, “Tabu search-part I”, *ORSA Journal on computing,* vol.1 Issue 3, pp. 190–206, 1989.

[4] J. B. Grimbleby, “Automatic analogue circuit synthesis using genetic algorithms”, *IEE Proceedings-Circuits, Devices and Systems,* vol. 147 Issue 6, pp. 319–323, 2000.

[5] E. Aarts, and K. Lenstra, “Local search in combinatorial optimization”, *Princeton: Princeton University Press,* 2003.

[6] F. T. S. Chan, and M. K. Tiwari, “Swarm Intelligence: focus on ant and particle swarm optimization”, *I-Tech Education and Publishing,* 2007.

[7] J. Kennedy, and R. C. Eberhart, “Particle swarm optimization”, *The IEEE International conference on neural networks, WA,* Australia, November 27–December 1, 1995.

[8] B. Benhala, A. Ahaitouf, A. Mechaqrane, B. Benlahbib, F. Abdi, E. Abarkan, and M. Fakhfakh, “Sizing of current conveyors by means of an ant colony optimization technique”, *IEEE Press, Proceedings of the 2nd International Conference on Multimedia Computing and Systems (ICMCS'11),* 7-9 April 2011, pp. 899-904.

[9] B. Benhala, A. Ahaitouf, M. Kotti, M. Fakhfakh, B. Benlahbib, A. Mecheqrane, M. Loulou, F. Abdi, and E. Abarkane, “Application of the ACO Technique to the Optimization of Analog Circuit Performances”, *Analog Circuits: Applications, Design and Performance,* pp. 235-255, Ed., Dr. Tlelo-Cuautle, NOVA Science Publishers Inc, 2011.

[10] E. Bonabeau, M. Dorigo and G. Theraulaz, “Inspiration for optimization from social insect behavior”, Vol. 406, pp. 39–42, 2000.

[11] M. Dorigo, G. DiCaro and L. M. Gambardella, “Ant algorithms for discrete optimization”, *Artificial Life Journal,* 5,137–172, 1999.

[12] M. Dorigo and S. Krzysztof, “An Introduction to Ant Colony Optimization”, *a chapter in Approximation Algorithms and Metaheuristics,* abook edited by T. F. Gonzalez, 2006.

[13] R.L. Haupt and S.E. Haupt, “Practical Genetic Algorithms”, *(book) John Wiley & Sons* 2004 , ISBN 0-471-45565-2.

[14] Goldberg, D. E., and Koza, J. R. (1990). “Genetic algorithms in search, optimization and machine learning”, *Workshop Notes, Computer Science Department, Stanford University,* August 6-10.