ISSN 2395-8618

Representation and Evaluation Mechanisms for Evolutionary Design of Analog Circuits

Aurora Torres Soto, María Dolores Torres Soto, Braulio Jesús Montoya Padilla,
and Eunice Ponce de León

***Abstract*—This work exposes the proposal of two essential elements when the automat-ed analog circuit design problem is approached by means of evolutionary computation: a representation and a fitness function mechanism. The analog circuit design problem is discussed, revealing in one hand, the need of a rep-presentation mechanism that exhibits the flexibility to adapt to very different topologies, remaining independent of the circuit´s complexity. And on the other hand, a fitness function that correctly leads the metaheuristics towards the building of a circuit with a specific behavior, while easily adapts to the representation mechanism. Subsequently a genetic algorithm is used to verify the efficiency of both mechanisms in combination with other routines in the design of one low-pass filter.**

***Index Terms*—Evolutionary computation, analog circuit design, representation mechanism, fitness function, low-pass filters.**

I. Introduction

Analog circuit design has been considered a craft activity, due to the enormous amount of knowledge, skills, and experience that it demands from the designer. The complexity of its nature has made it the target of the development of multiple tools. This kind of tools has been created with the purpose to help engineers in the stages that are involved to achieve an effective and efficient device. The quality in the design of an electrical circuit can be described according to design time, behavior of the final circuit, number of elements it has, the power it dissipates, among others.

Nowadays, design support tools present an interesting range of solutions that support the domain of digital circuit design, however, the scene in relation to analog circuit design is very different. Although it is a fact that since the 70s many of the analog circuits have been replaced by digital ones, many important functions remain analog due to the analog nature of transducers [1]. Adding to the above the fact that analog design is intensive, time consuming, and requires specialized knowledge and multiple skills [2], it is not surprising that multiple efforts are being made to support it.

The design of electronic devices requires a functionally correct circuit, as simple and efficient as possible, and in the shortest time span. The use of simulation tools has substantially aided the design process, since they allow different analyses to be carried out on the circuits to verify that they exhibit the expected behavior prior to manufacturing them.

The enormous success that evolutionary algorithms have had in the digital circuits design [3,4,5,6] and among others, and even in areas outside engineering design, such as medicine [7,8] and urban planning [9] to mention a few; has strongly motivated the use of these tools in complex scenarios, such as the one we are dealing with.

The present work proposes to approach the analogical design using evolutionary computation, which implies, at least a representation mechanism and an objective function as already mentioned [10]. With the above idea in mind, representation and evaluation mechanisms designed to be employed by an evolutionary computation tool are presented and discussed.

II. Preliminaries

A. The Analog Circuit Design Problem

Synthesis of circuits is the process of conceptualizing an electronic device that must meet certain specifications established in advance. Two basic stages must be taken into consideration in this process: a) establishment of the connections among the circuit elements *(“topology”)* and b) determination of the type and value of each element *(“sizing”)* [11].

Regarding the establishment of the specifications of a circuit, it is the step in which the design problem is defined. Most of the research related with analog circuit design propose the use of fixed topologies as a starting point and subsequently, the establishment of the values of the circuit elements using deterministic, stochastic and hybrid methods, as can be seen in [12, 13, 14].

Even with convenient and fixed topologies, it is common that this may not lead to good results and the optimization process must start from the beginning with a new topology [15]. This scenario confirms that the automatic synthesis of *topology* of analog and radio frequency circuits is currently an open problem as established by Sorkhabi and Zhang [16]. In addition, the computational cost involved in building new libraries for each new circuit; has caused the scientific community to look for other alternatives.

The automatic generation of the *topology* refers to the establishment of the way in which the elements that form a circuit are going to be connected starting from a “blank sheet” [17]; approach considered in this work.

B. Evolutionary Computation

From a simple point of view, natural evolution is an optimization process in which the objective is to improve the ability of organisms or systems to survive in a dynamic and competitive environment [18].

Because evolution behaves as a process of species improvement, scientific community has been motivated to

Manuscript received 20/05/2019, accepted for publication on 17/11/2019.

The authors are with the Autonomous University of Aguascalientes, Mexico ({atorres, mdtorres, eponce}@correo.uaa.mx, al188428@edu.uaa.mx).

https://doi.org/10.17562/PB-63(2)-3

ISSN 2395-8618

```
General Evolutionary Algorithm

1. Let G=0 generation counter
2. Create and initialize an n-dimensional population P(0)
3. Repeat
    • Evaluate the fitness f(x), of each individual x, in the population P(G)
    • Produce offspring
    • Select the population P (G + 1) of the new generation
    • Advance to the next generation. G = G + 1
4. Until the stop condition is met
```

Fig. 1. General pattern of an evolutionary algorithm

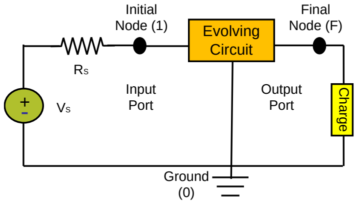

Fig. 2. Template for a two-port electrical network

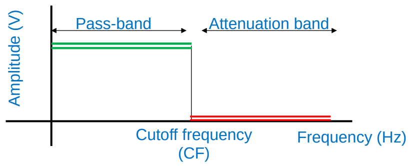

Fig. 3. Frequency response of an ideal filter

adopt some of its mechanisms in solving problems. Evolutionary computation is the area of computer sciences that uses computational models of the evolutionary process, such as natural selection and the survival and reproduction of the fittest individuals, to solve computer-based problems [18]. Although there is a wide variety of evolutionary algorithms, they all have in common that implement an evolutionary process to build a search or optimization algorithm. Figure 1 summarizes the most relevant aspects of an evolutionary algorithm; however, it should be mentioned that some paradigms may vary from this pattern.

Some of the main paradigms on evolutionary computing are listed below:

1. Genetic Algorithms [19, 20],
2. Genetic Programming [21],
3. Evolutionary Programming [22],
4. Estimation of the Distribution Algorithms [23],
5. Evolutionary Strategies [24],
6. Differential Evolution[25].

Because the Genetic Algorithm (GA) was the first that reported practical results, it is the most popular exponent of evolutionary computation. For this reason, the mechanisms developed and presented in this paper are manipulated by a GA; however, they can be adapted by any technique of evolutionary computation.

III. Mechanisms for Evolutionary Design of Analog Circuits

All the tools for generating *topology* use as a starting point a template to which the elements are connected [17]. In the case of two-port electrical networks (where passive filters belong), it is frequent that the template has two nodes that represent the input port and the output port of the circuit.

The template used in this work is shown in Figure 2. This pattern has been used in the evolution of different types of filters [23,25,27,29].

Figure 2 shows that the input port of the electrical network is located between initial node (1) and ground (0), while the output port is located between the final node (F) and ground (0). Both the process of generating *topology*, as well as that of determining the values of the elements (*sizing*), take place between the three mentioned nodes. Naturally, this template can be changed according to the design objective.

For this work, the representation mechanism is analyzed assuming that the design objective is a passive analog filter, a circuit built exclusively of passive circuit elements. This type of electrical network is the one that has been most frequently used by researchers in the area since filtering is a well understood discipline [28].

An electronic filter can be viewed as a frequency-selective device, allowing the magnitude or phase response to be shaped in a prescribed way [30]. In other words, it is a circuit that blocks signals of a certain range of frequencies, while allowing those with frequencies in a different range to pass [31].

Figure 3 shows what is known as frequency response of an ideal low-pass filter. This filter, according to its name, allows the electrical signal to pass when its frequency is below the cutoff frequency (CF), completely attenuating those whose frequency is higher than CF.

It is worth mentioning that the response described in the previous figure corresponds to an ideal theoretical model, but using real devices, it is not feasible to obtain this behavior; however, the quality of the filter is related to the fidelity of its behavior in relation to the ideal. To measure the functionality of electronic filters, they are exposed to different frequency conditions of the power source, while the voltage level at the output is evaluated; however, this aspect will be discussed later in section 3.2 of this paper.

C. *Representation Mechanism*

To use an evolutionary tool for the solution of a complex problem, the first step is to establish the representation of a possible solution to the problem. The mechanism used to represent an individual (solution), is known as "chromosome".

An efficient representation is essential for the optimization process to be successful, and it should exhibit the following characteristics [17]:

— Flexibility: Encoding adaptable enough to represent a wide variety of individuals.

— Simplicity: Decoding as easy as possible.

— Robustness: Ability to withstand the modifications carried out by evolvable operators.

Considering that the evolution should affect both: *topology*, and *sizing*, it was created a representation that allows the

ISSN 2395-8618

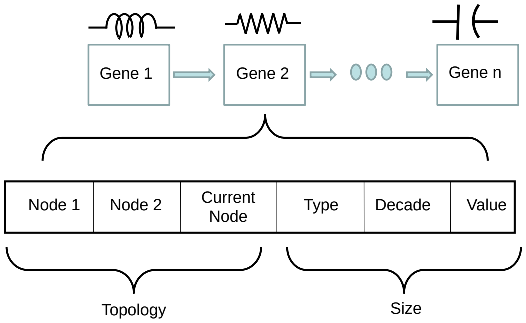

Fig. 4. Chromosome and description of a gene

TABLE I
SERIES OF COMMERCIAL VALUES

Values	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
E12	10	12	15	18	22	27	33	39	47	56	68	82
E6	10	15	22	33	47	68						

TABLE II
PROPOSED CODING FOR PASSIVE CIRCUIT ELEMENTS

Type	**Decade**	**Value**
C (0)	$10^{-5} – 10^{-9}$ (0 – 4)	E6 (0 – 5)
R (1)	$10^{1} – 10^{6}$ (0 – 5)	E12 (0 – 11)
L (2)	$10^{-1} – 10^{-6}$ (0 – 5)	E12 (0 – 11)

genetic operators to manipulate both aspects. According to the classification of Mattiussi and Floreano [32], this mechanism is known *as development representation technique*. Although this kind of mechanisms is more complicated than the *direct representation techniques*; they are more compact although they require a mechanism to guide the development process (network construction).

The mechanism for generating possible solutions to this problem is beyond the scope of this paper but interested readers can find a detailed description of it in [17].

A circuit is represented by variable length chromosomes, this way no expert knowledge of the problem is required, and the exploration potential of evolutionary algorithms is not limited [33]. A chromosome (candidate solution or circuit) is represented by a linked list of nodes (genes); where each node represents a circuit element with information of its *topology* and its *sizing*. Figure 4 shows the chromosome designed for the evolution of passive analog circuits. It presents the way a chromosome is made up of genes and the type of information that each gene stores according with Torres et al. [34].

As shown in the figure above, with each circuit element, information describing its connections, type and dimension is stored.

To facilitate the implementation of the resulting circuit and reduce the gap between the evolved and the implemented circuit; the circuit representation mechanism includes the concept of commercial values proposed by Khalifa and Horrocks [35]. The values that each circuit element can take depend on the type of element. To guarantee the use of commercially available values, the series of values known as E6 and E12 are used.

The first three fields of the node establish the connections (*topology*) while the last three describe the type of circuit element and their value (*sizing)*. The value of each circuit element is set using equation 1:

$$cvalue * 10^{cdecade}, \qquad (1)$$

where:

$cvalue$ is the encoded value according to the series E6 or E12 as it corresponds to it (see table 1).

$cdecade$ is the encoded exponent in the decade field of the gene according to the type of circuit element (see table 2).

Table 1 shows the normalized values of electronic components according to the E12 and E6 series. According to the commercial values of the series that are being used, the value of the device will correspond to the value of the series; for example, if the value field has a 2 and the series E12 is being used, the encoded value to be taken is 15; while if it is the E6 series, a value of 2 corresponds to 22. It is worth mentioning that although only the most common series are used, the coding can be used for series such as E24, E48, or any other.

The range of values that an element can take, depends on its type. For example, typical practical values of resistors range from KΩ to MΩ; while typical values for capacitors are below µF and inductances typically exhibit values on the order of mH. Keeping these practical limitations in mind; a range of values that last three fields of the chromosome can take, are shown in Table 2. Although it should be mentioned that there is no impediment to expand or to restrict this range according to the type of circuit to be designed.

Thus, to represent a 100KΩ resistor through the E12 series, we must use Type = 1, Decade = 3 and Value = 0 ($10 \text{x} 10^{4}$ which is equivalent to 100000 ohms).

ISSN 2395-8618

TABLE III
SOME OUTPUT VOLTAGES VERSUS INPUT FREQUENCY VARIATIONS

Frequency (Hz)	**Voltage gain (dB)**	**Frequency (Hz)**	**Voltage gain (dB)**
1.000	0	8.923K	-13.12
1.614	0	13.34K	-16.57
2.371	0	20.27K	-20.07

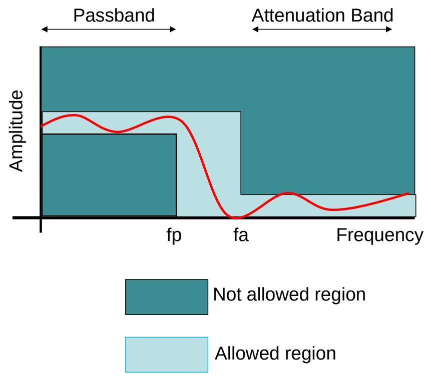

Fig. 5. Frequency response of a real low-pass filter

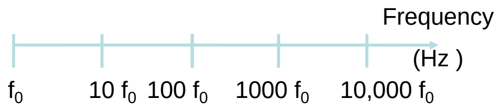

Fig. 6. Circuit evaluation frequency band

Regarding the coding of *topology* of the electrical network, it is found in the first fields of the chromosome: *Node 1* corresponds to the node to which terminal 1 of the circuit element is connected and *Node 2* corresponds to the node of the network to which terminal 2 of the device is connected. *Current Node* is a pointer to the node the next element of the network should be connected. The purpose of this pointer is to take care that the electrical network does not remain open somewhere. Different examples of the use of this coding and their explanation can be found in [16].

D. Evaluation Mechanism

The ability of the evolutionary process to find good solutions to a specific problem depends very much on the efficiency of the adaptability function, which drives the search process. So, the establishment of the fitness function of the circuits must be able to evaluate their performance. In the case of filters, it is possible to establish a measure of their behavior if they are subjected to different frequency conditions of the power source, while the voltage level at the output is evaluated. This procedure is known as frequency analysis and let it know the frequency response of the circuit.

If the amplitude of a voltage generator is adjusted to *"V"* volts and its frequency to *"F"* Hz, while measuring the voltage *"W"* volts at its output, the *V/W* ratio for each frequency is called the *"frequency response"* of a circuit, and it allows to evaluate the circuit performance. The frequency response information is typically presented in terms of voltage gain *"Hv"* according to equation 2 and measured in decibels [dB]:

$$Hv = 20log\left(\frac{W}{V}\right). \tag{2}$$

Table 3 shows a sample of series of voltage gain values obtained when holding the amplitude of the input signal at a fixed level, while modifying the source frequency in a simple low-pass filter; whose cutoff frequency is approximately 2KHz.

The proposed objective function grows as a larger number of frequency points are tested.

The frequency response of a real low-pass filter, used to test the proposed evaluation mechanism, is shown in figure 5. In this figure the frequencies *fp* and *fa* represent the frequency at which the passband ends and the frequency at which the attenuation band begins, respectively.

To guarantee the good performance of a filter in a wide range of frequencies, its evaluation must include frequencies below and above the mentioned critical frequencies.

The proposed evaluation function assumes the rating of "P" points per decade on a logarithmic scale. This scale begins at an initial frequency *"f_0"* and ends at *"D"* decades later. The scale in figure 6 has 4 decades. This way, if 20 points are sampled per decade then 81 different frequency conditions should be evaluated.

This situation is the reason why the evaluation of the objective function in circuit design problems is so time and computational resources consuming. The time that the evaluation of the objective function takes in this type of problem is such that some researchers have chosen to run this part of the problem in parallel way; for example, Koza et al. [21] used 64-microprocessor equipment (Power PC); while Lohn et al. [26] reported have used a cluster of 6 Sun Ultra workstations.

To be able to work on this kind of problems in a less sophisticated computational equipment, it is important not only to keep the number of circuits that will not produce results as low as possible; but also simplify the translation procedure among their representation and the circuit simulation software used to calculate its adaptability.

To measure the proximity between a circuit and the desired solution, and consequently to continue improving solutions from a generation to another, a reliable fitness function is essential. The fitness function proposed in this work is a measure of the error "ε" between the f*requency response* of the target filter and the response of the evolving filter at a certain frequency "f_i".

According to Equation 3, "ε_i" is the absolute value of the deviation between the magnitude of the response of the target filter $M(f_i)_{Obj}$ and the magnitude of the response of the evolving filter $M(f_i)_{Evol}$ at the frequency f_i:

$$\varepsilon_i = \left|M(f_i)_{Obj} - M(f_i)_{Evol})\right|, \tag{3}$$

where:

$M(f_i)_{Obj}$ and $M(f_i)_{Evol}$: are voltage values calculated at the output of the target circuit and the evolving circuit respectively.

f_i: Is the i-th frequency point.

The measurement of the distance ε_i is calculated at each of the *N* frequency sampling points *(N = PxD);* and since the

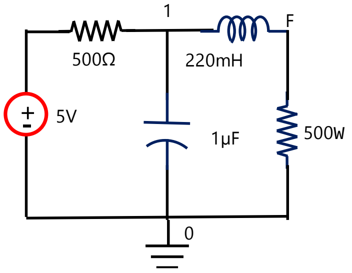

Fig. 7. Low pass filter with CF=750Hz

response of the filter being designed is limited to the allowed region (see Figure 5), a penalty factor *"λ"* has also been used; this *"λ"* increases the value of the difference of those points whose response touches the not allowed zone.

Equation 4 shows the function that measures the discrepancy between the objective filter and the one evaluated, taking *N* frequency points and introducing the mentioned penalty factor. Equation 5 represents the normalized evaluation function:

$$\xi = \sum_{i=1}^{N} \lambda(\mathcal{E}_i) * \mathcal{E}_i \,, \tag{4}$$

$$F = \frac{1}{1+\xi} . \tag{5}$$

According to Equation 5, the evaluation function should take the value of one when the difference between the objective filter and the evolved is zero *(ξ=0)*. Nevertheless, when the filter being evaluated is completely different from the target, the function will take a value that depends on the filter specifications and the value taken by the penalty factor *"λ"*.

An example of a coded circuit obtained using the fitness function described above is shown in figure 7. This circuit allows the transmission of electrical signals whose frequency is below 750 Hz and blocks signals whose frequency is above 750 Hz. This circuit obtained an evaluation of 0,7948 by means of a genetic algorithm.

Although the implementation aspects of the genetic algorithm are beyond the scope of this paper, it is worth mentioning that it was run with a population size of 50 circuits for 5 generations. Making use of a crossover probability of 90% and a mutation probability of 30%.

IV. Conclusions

Because two of the main challenges in the automatic design of analog circuits are coding and evaluation, in this work a representation mechanism that was created bearing in mind that it must be capable of evolving both *topology* and *sizing* of circuits is proposed. On the other hand, also is proposed a robust evaluation mechanism, designed to be used in an evolutionary algorithm. A good performance of the circuit will be rewarded with a higher fitness value and a bad circuit will have a lower value.

The representation mechanism, in addition to being very compact, since it is totally independent of the complexity of the circuit topology, encodes a single gene for each circuit element. This situation produces an interesting contribution in relation to other proposals, because although they are not discussed in this document, the transformation operators explored for their implementation produce only valid topologies, which greatly reduces the computational load compared with other schemes such as Koza's [21]. The chromosome is of variable size and is computationally implemented, using lists of linked nodes (genes), which exploits the search potential of evolutionary algorithms.

In this work it is shown that the proposed representation mechanism adapts to the suitable circuit template, depending on the type of electronic device that will be designed and evaluated. An aspect that is not yet discussed in this work, but that has already been explored, is the easiness of converting coded circuits to their equivalents in circuit description language, for their simulation.

Another advantage of this mechanism is that it is adaptable for the manipulation of a large group of families of electronic devices, which reduces gap between an evolved circuit and an implemented one. In relation to the handling of different scales of devices, our proposal is very flexible.

Additionally, an evaluation mechanism that is easy to understand, intuitive and that guarantees the proper performance of the circuit is also proposed. Because it involves simulating the circuit, it allows the result to be more reliable. It also is a mechanism that adapts easily to several evolutionary algorithms.

Finally, in this work the version for the representation of passive analog elements is shown, but the version that integrates active elements is already being analyzed.

In this work, aspects of the execution of evolutionary algorithms are not discussed yet, since its purpose is to present the representation mechanism, in combination with the evaluation mechanism, which can be coupled to evolutionary algorithms in a direct and efficient way.

References

[1] F. Castejón and E. J. Carmona, "Automatic design of analog electronic circuits using grammatical evolution," *Applied Soft Computing*, vol. 62, pp. 1003–1018, 2018. DOI:10.1016/j.asoc.2017.09.036.

[2] Ž. Rojec, A. Bűrmen and I. Fajfar, "Analog circuit topology synthesis by means of evolutionary computation," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 48–65, 2019. DOI:10.1016/j.engappai.2019.01.012.

[3] C. Coello, D. Ectrica and S. Computacion, "Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization," in *Proceedings The Second NASA/DoD Workshop on Evolvable Hardware*, 2003. DOI:10.1109/EH.2000.869354.

[4] U. R. Karpuzcu, "Automatic Verilog Code Generation through Grammatical Evolution", in *Genetic and Evolutionary Computation Conference, GECCO*, 2005. DOI:10.1145/1102256.1102346.

[5] X. Yan, W. Li, Y. Zhang and J. Wu, "Electronic Circuit Automatic Design Based on Genetic Algorithms," *Procedia Engineering*, vol. 15, pp. 2948–2954, 2011. DOI:10.1016/j.proeng.2011.08.555.

[6] M. Anjomshoa, A. Mahani and S. Sadeghifard, "A new automated design and optimization method of CMOS logic circuits based on Modified Imperialistic Competitive

Algorithm," *Applied Soft Computing Journal*, vol. 21, pp. 423– 432, 2014. DOI:10.1016/j.asoc.2014.04.011.

[7] M. D. Torres, A. Torres, J. C. Aguilar et al., "Máquinas de soporte vectorial para pronóstico en la osteosíntesis de la fractura transtrocantérica de cadera," *Research in Computing Science*, vol. 149, no. 8, pp. 985–996, 2020.

[8] A. E. G. Acosta, M. D. T. Soto, A. T. Soto and E. E. P. de León Sentí, "Contrastación de algoritmos de aprendizaje automático para la clasificación de señales EEG" *Research in Computing Science,* vol. 149, pp. 515–525, 2020.

[9] P. R. de León, M. D. T. Soto and A. T. Soto, "Mecanismo de clasificación para diabetes mellitus en la población de Aguascalientes", *Research in Computing Science,* vol. 149, pp. 527–540, 2020.

[10] C. Coello et al., "Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach," *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 16, no. 1, 2003. DOI:10.1017/ S0890060401020054.

[11] A. Das and R. Vemuri, "A graph grammar based approach to automated multi-objective analog circuit design", in *Proceedings of the Conference: Design, Automation & Test in Europe,* pp. 700–705, 2009. DOI:10.1109/DATE.2009.5090755.

[12] A. el Dor, M. Fakhfakh and P. Siarry, "Multiobjective Differential Evolution Algorithm using Crowding Distance for the Optimal Design of Analog Circuits," *Journal of Electrical Systems*, vol. 12, no. 3, pp. 612–622, 2016.

[13] A. C. Sanabria-Borbón and E. Tlelo-Cuautle, "Sizing analog integrated circuits by combining gm/ID technique and evolutionary algorithms," in *IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS),* pp. 234–237, 2014. DOI:10.1109/MWSCAS.2014.6908395.

[14] J. Olenšek, T. Tuma, J. Puhan and A. Bűrmen, "A new asynchronous parallel global optimization method based on simulated annealing and differential evolution," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1481–1489, 2011. DOI:10.1016/j.asoc.2010.04.019.

[15] G. Shi, "Toward automated reasoning for analog IC design by symbolic computation–A survey," *Integration, the VLSI Journal*, vol. 60, pp. 117–131, 2018. DOI:10.1016/ j.vlsi.2017.08.005.

[16] S. E. Sorkhabi and L. Zhang, "Automated topology synthesis of analog and RF integrated circuits: A survey," *Integration, the VLSI Journal,* vol. 56, pp. 128–138, 2017. DOI:10.1016/ j.vlsi.2016.10.017.

[17] A. Torres-Soto, "Metaheurísticas Evolutivas en el Diseño de Circuitos Analógicos," Benemérita Universidad Autónoma de Aguascalientes, 2010.

[18] K. Meng, Z. Y. Dong and Y. Qiao, "Swarm Intelligence in Power System Planning," *International Journal of Clean Coal and Energy*, vol. 2, no. 2, 2013. DOI:10.4236/ijcce.2013.22B001.

[19] J. H. Holland, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence," MIT Press, Cambridge, MA, USA, 1992.

[20] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning*, vol. 3, pp. 95–99, 1988. DOI:10.1023/A:1022602019183.

[21] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, no. 4, pp. 87–112, 1994. DOI:10.1007/BF00175355.

[22] L. J. Fogel, "On_the_Organization_of_Intellect," Ph.D. Thesis, University of California, Los Angeles, 1964.

[23] P. Larrañaga and J. A. Lozano, "Estimation of Distribution Algorithms," Springer US, Boston, MA, 2002.

[24] H. P. Schwefel, "Evolutions strategie und numerische Optimierung," Springer, 1975.

[25] F. Vitaliy, "Differential Evolution," Springer US, Boston, MA, 2006.

[26] J. D. Lohn and S. P. Colombano, "A circuit representation technique for automated circuit design". *IEEE Transactions on Evolutionary Computation,* vol. 3, no. 3, 1999. DOI:10.1109/ 4235.788491.

[27] J. D. Lohn and S. P. Colombano, "Automated analog circuit synthesis using a linear representation," in *International Conference on Evolvable Systems,* 1998.

[28] C. Goh and Y. Li, "GA automated design and synthesis of analog circuits with practical constraints," in *Proceedings of the 2001 Congress on Evolutionary Computation,* 2001. DOI:10.1109/ CEC.2001.934386.

[29] J. Hu, X. Zhong and E. D. Goodman, "Open-ended robust design of analog filters using genetic programming," in *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO'05*, ACM Press, New York, USA, 2005. DOI: 10.1145/1068009.1068283.

[30] A. Ambardar, E. Urbina, G. Nagore and L. Hernández, "Procesamiento de señales analógicas y digitales," Thomson Learning, México, 2002.

[31] C. D. Johnson, "Process control instrumentation technology".

[32] C. Mattiussi and D. Floreano, "Analog Genetic Encoding for the Evolution of Circuits and Networks," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 5, 2007. DOI: 10.1109/TEVC.2006.886801.

[33] R. S. Zebulum, M. Vellasco and M. A. Pacheco, "Variable Length Representation in Evolutionary Electronics," *Evolutionary Computation*, vol. 8, no. 1, 2000. DOI:10.1162/ 106365600568112.

[34] A. Torres, M. Torres and L. E. P. de Ponce, "Mecanismo de representación para la evolución automática de circuitos analógicos Representation Mechanism for Automatic Evolution of Analog Circuits," *Research in Computing Science*, vol. 149, no. 253–265, 2020.

[35] Y. Khalifa, D. H. Horrocks and Y. M. A. Khalifa, "Genetically derived filter circuits using preferred value components," in *IEEE Colloquium on Analogue Signal Processing,* 2002.