Audio Circuits Evolution through Genetic Algorithms

P. H. G. Coelho, J. F. M. do Amaral, E. N. Da Rocha and M. C. Bentes
State Univ. of Rio de Janeiro, FEN/DETEL, R. S. Francisco Xavier, 524/Sala 5001E, Maracanã, RJ, 20550-900, Brazil

Keywords: Genetic Algorithms, Artificial Intelligence Applications, Evolutionary Electronics.

Abstract: This paper focuses on the implementation of an extrinsic platform in order to develop audio electronic circuits with known topologies. This platform uses genetic algorithms to choose the best components to achieve a specific goal. All the proposed topologies have been consolidated in the audio market and due to this aspect, the technique has not proposed new possibilities for them. However, the generation mechanism of these alternatives is studied in depth, approaching their theoretical potential. User specifications of the proposed interface define the evolution of the circuit. The developed platform compares the graphic of a simulation with a graphic which is generated by a function in MATLAB. The similarity between these curves is used as the fitness function to evolve the circuit components values. All the work was done in MATLAB and Simulink. As MATLAB is used to run the codes and create desired curves with its function, its tool Simulink is used to simulate circuits and to carry out transfer functions analysis. Case studies are presented to illustrate the method which are analogic filters used in audio applications. These circuits were evolved by using the free package GAOT for MATLAB.

1 INTRODUCTION

Evolutionary projects are proposals for global optimization and can be understood as search procedures for optimal solutions, in order to design a project. Artificial intelligence is usually inspired by nature for modeling algorithms. Examples are neural networks based on brain behavior, genetic algorithms inspired on biological evolution and other approaches found in works of researchers in the field of artificial intelligence. For instance, the Gray Wolf Optimizer (GWO) proposed by (Mirjalili et al., 2020) is a stochastic algorithm based on the hunting behavior of a pack of gray wolves. This paper will address applications through the use of genetic algorithms, whose properties will be used in the field of evolutionary electronics. Its operations have little complexity, and its structure has very well-defined, easy-to-understand steps. The act of programming, in general, has been inspired by nature even for naming its terms. Artificial intelligence is no exception for this aspect. For example: Object-oriented languages work with the concept of inheritance, that is, a child class can inherit certain attributes from the parent class, as desired by the coder. It is also common for these terms to appear in articles e.g., the Heapsort sorting algorithm, (Sedgewick, 1998) which uses parent and child elements. Evolutionary electronics purpose is a new paradigm for problem solving inspired by Natural Selection as proposed by Darwin, conceived after his observations carried out in the Galapagos Islands. From the observation of birds with different beaks on these islands, the scientist concluded that the birds should have a common ancestor, but would have evolved in different environments: those that underwent beneficial mutations for their environment survived and managed to pass their traits on to their descendants (Coello Coello, 1999) (Coello Coello, 2013). It was proposed that species could always change over time, and that new species would emerge from pre-existing species, all of which would share a common ancestor. In this model, each species has a unique set of heritable genetic differences compared to the common ancestor, which gradually accumulated over time. Repeated events of differentiation, in which new species diverge from their common ancestor, produce a multilevel "tree" that connects all living things, known as Darwin's tree for life. When developing an evolutionary platform, the practitioner must choose one of three implementation methods: intrinsic, extrinsic, or hybrid (Greenwood and Tyrrel, 2007), (Reorda et. al., 2017). In the intrinsic

Coelho, P., M. do Amaral, J., N. Da Rocha, E. and Bentes, M.
Audio Circuits Evolution through Genetic Algorithms.
DOI: 10.5220/0011062500003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 514-520
ISBN: 978-989-758-569-2; ISSN: 2184-4992

evolutionary platform, the circuit is designed in hardware by the genetic algorithm. The implementation also takes place in hardware. In the extrinsic evolutionary platform this entire process takes place in software. The hybrid platform mixes the concept of both. The first method is better in terms of the possibility of also being able to verify the circuit's operation physically, while the second only addresses the operation through a simulator. This work will follow the second option and presents the implementation of an extrinsic platform, in order to enable the evolution of usual audio electronic circuits. The modeling presented is based on a genetic algorithm for evaluating the choices of components that approximate the circuit's operation to the desired specification. This paper is organized in four sections. The second section describes the basics of evolutionary environment and the proposed platform. Section three discusses case studies in connection with the evolutionary circuits' platform. Finally, section four ends the paper with the conclusions.

2 EVOLUTIONARY PLATFORM

2.1 Extrinsic Platform

The extrinsic platform (Coelho et. al., 2021) is carried out through the use of circuit simulators, with implementation only in software. It gives the designer more freedom to find more varied solutions regarding topology. There is no limitation on the types and components that can be used in the population. However, extrinsic evolution will require a much longer time compared to its intrinsic equivalent, requiring high processing capacity of the machine on which the genetic algorithm and simulator will be executed. Figure 1 shows the working diagram of this type of platform.

The simulator brings to the process the ability to establish a specific configuration for calculations of some regime parameters such as noise, temperature and others, and thus generate candidates even closer to the desired ones. The developed platform focuses on analyzing the numerical values generated by the genetic algorithm directly within MATLAB, which was used to trigger Simulink within the development environment itself, to simulate the circuit with the solutions proposed by the genetic algorithm and also to obtain functions circuit transfer. MATLAB has several toolboxes, some important for the area of artificial intelligence, such as neural networks, genetic algorithms and fuzzy logic. There is a toolbox named GAOT (Genetic Algorithm Optimization Toolbox) whose algorithm was developed by researchers at North Caroline State University, so that it could be used directly within the MATLAB development environment.

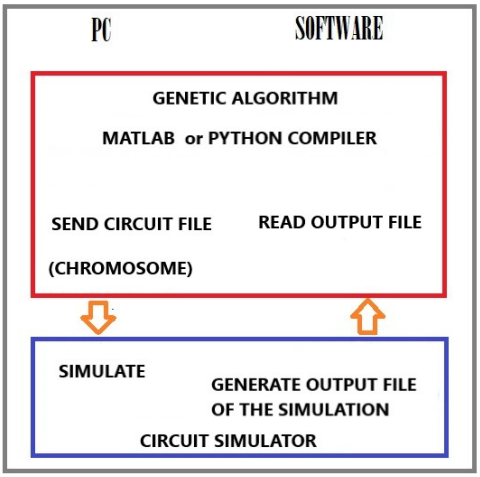

Figure 1: Extrinsic platform.

This toolbox was used in this work for several reasons, the main one is that it has open code, without any restrictions for changes. It should be noted that MATLAB has an optimization function based on genetic algorithms, the GA function, which comes with the software. Figure 2 shows a flowchart of the developed evolution platform, based on a genetic algorithm, allowing the transfer function of a circuit to be obtained, and also the analysis of any system through this function. Simulink has two functions within the platform. It is used to obtain circuit transfer functions, at the time of modeling, in which the platform user does not know the mathematical description of the relationship between the output and input of the circuit. In addition, it can be used as a simulator after obtaining the optimal values. In order to make the platform's operation clearer, figure 3 presents a brief description of the evolution process. First, the circuit topology is chosen for the evolution of the component values. Next, their ranges of values are related to the genes of the individuals in the genetic algorithm, thus determining the size of the chromosomes. After these steps, the parameters for executing the genetic algorithm must be defined. This configuration will depend on the designer's objective as well as the design specifications. Finally, the solution is analyzed using the best chromosome, i.e. the transfer function obtained for the optimal solution is compared with the one considered ideal, that is, the one conceived by a MATLAB function. The

comparison match is a guarantee that the evolved circuit meets all circuit specifications.

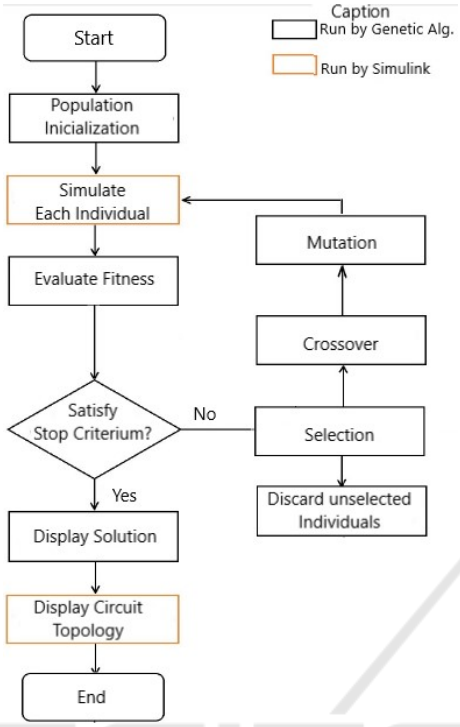

Figure 2: Flowchart of the evolutionary platform using GAOT.

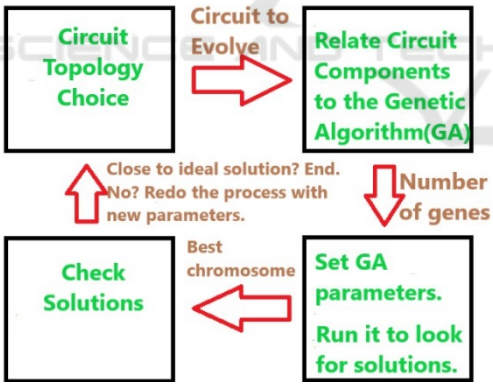

Figure 3: Evolution process through the extrinsic platform.

2.2 Platform Facts

The extrinsic platform development steps are then:

1. Choose the topology for which an optimal solution is sought.
2. Describe the behavior of the circuit that comprises the modeling of its frequency response.
3. Calculate the error.
4. Obtain the desired evolved circuit.

The choice of the fitness function in order to assess the evolutionary adequacy of the platform is given by equations 1 and 2.

$$Fitness = \frac{1}{1+error} \quad (1)$$

$$error = \sum_{i=1}^{N} \left| \frac{Output_{Goal}(i) - Output_{Observed}(i)}{N} \right| \quad (2)$$

3 CASE STUDIES

The developed extrinsic platform is capable of synthesizing a circuit and presenting an optimal solution, given the specification and topology of a circuit. In order to analyze its efficiency, several case studies were developed for the analysis of the electronic evolutionary process, of which two were selected. The first case addresses a notch filter for which an optimal solution was sought that would meet an attenuation at a specific frequency. The second case study is a low pass filter, aiming to meet a first order active filter.

3.1 Case Study 1: Notch Filter

Notch filters, also known as band-rejectors, are conceived in their idealization through the superposition of a high-pass filter and a low-pass filter. The notch filter used is based on the topology shown in Figure 4.

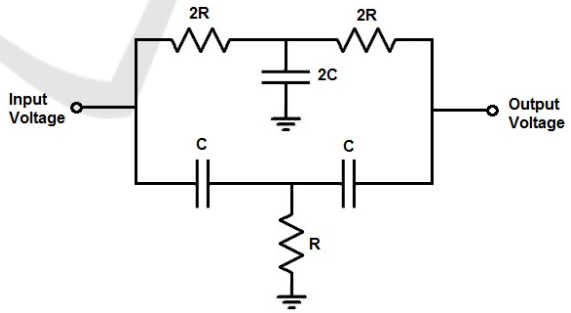

Figure 4: Notch filter basic circuit.

The evolution parameters are:

- Number of generations: 100
- Number of individuals per generation: 100
- Crossover Probability: 0.85
- Probability of mutation: 0.001

Circuit specifications are given in table 1.

Table 1: Range of resistor and capacitor values for simplified notch filter topology.

Parameters	Range of Values
Resistor	10 kΩ - 100 kΩ
Capacitor	100 nF - 700 nF
Attenuated Bandwidth	2 Hz
Attenuation frequency	60 Hz

Figure 5 shows the evolution of fitness for individuals in the population from the genetic algorithm to the notch filter.

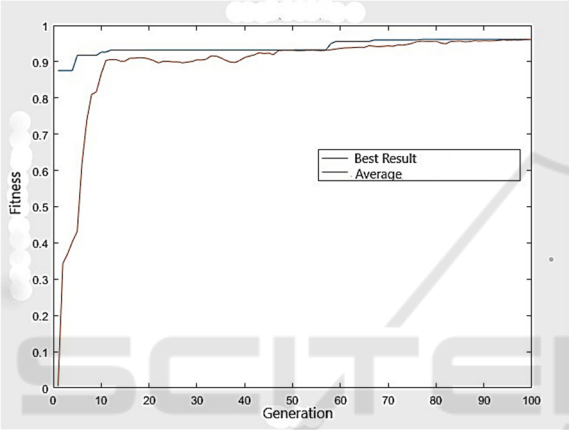

Figure 5: Notch filter evolution.

The fitness curve starts from a relatively high value (approximately 0.85) and quickly converges to 0.95. The average fitness converges to a value close to 0.95 as well. It is possible to simulate the circuit in Simulink or any external simulator. The evolved notch filter including the values proposed by the genetic algorithm is shown in Figure 6. The proposed platform does not work with components commercial values. Therefore, it is necessary to adapt the values obtained, either considering the loss of efficiency, exchanging for the equivalent of a close value, or associating resistors and capacitors to be as close as possible to the solution proposed by the genetic algorithm. After obtaining the optimal values, it is possible to simulate the circuit in Simulink or any other external simulator. Therefore, the response curve of the circuit evolved by the genetic algorithm on the platform and the target response curve can be compared.

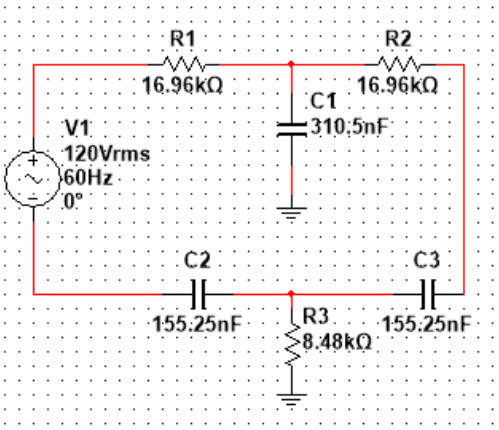

Figure 6: Notch filter evolved circuit.

Figures 7 and 8 show the graphs of the Bode frequency response in magnitude and phase of the evolved curve and that of the target, confirming the high degree of similarity between them.

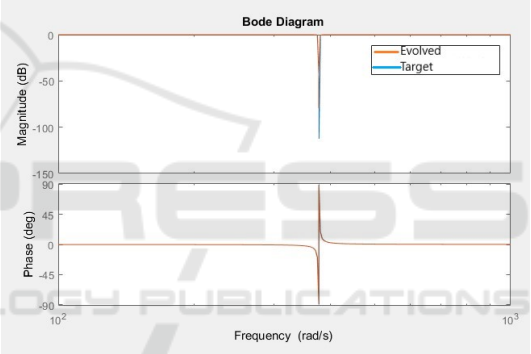

Figure 7: Notch filter frequency response for target and evolved circuit.

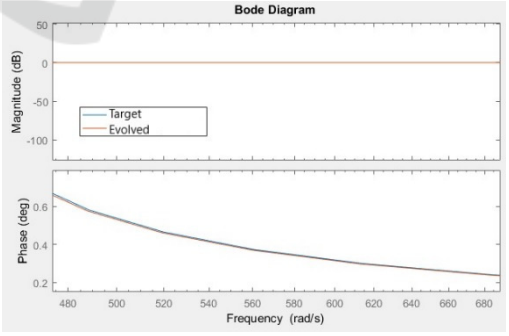

Figure 8: Notch filter in the region after the rejection band.

3.2 Case Study 2: Low Pass Filter

The low pass filter chosen for the evolution was an active filter, composed of op-amp, resistors and capacitor whose basic circuit is shown in figure 9.

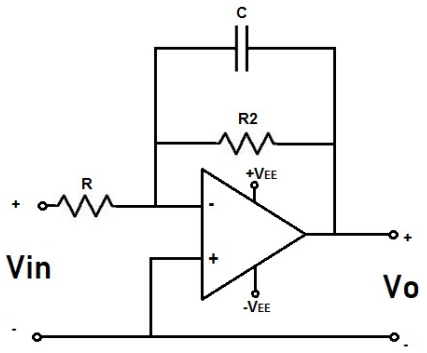

Figure 9: Low pass filter basic circuit.

The filter specifications are shown in Table 2.

Table 2: Low pass filter specifications.

Parameters	Range of Values
Resistor	10 kΩ - 100 kΩ
Capacitor	0.01 nF - 10 nF
Gain	2
Quality Factor (Q)	4
Cut-off Frequency	200 Hz

The evolution parameters considered for the present case are:

- Number of generations: 100
- Number of individuals per generation: 30
- Crossover Probability: 0.95
- Probability of mutation: 0.001

Figure 10 shows the evolution of fitness for the low pass filter case study for the best chromosome. It is noted that, unlike the graph generated for the notch filter case, the first generations would not provide good solutions for the circuit. The evolution of each generation in relation to the previous one was more significant. Note also that, again, the maximum fitness stabilized at a value close to 0.95. However, this value was reached with fewer generations than in the previous case, and, again, the convergence of the algorithm was fast. Table 3 summarize satisfactory results obtained for the evolution of the low pass filter.

Table 3: Summary results for the low pass filter.

Parameter	Value	Error
Gain	2.09	0.09
Cut-off Frequency	196 Hz	4 Hz

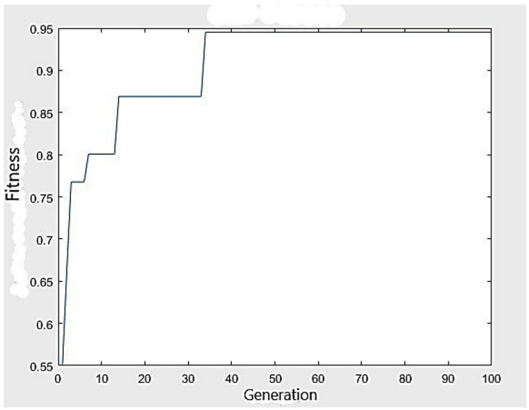

Figure 10: Low pass filter evolution best result.

The evolved low pass filter including the values generated by the extrinsic platform is shown in Figure 11.

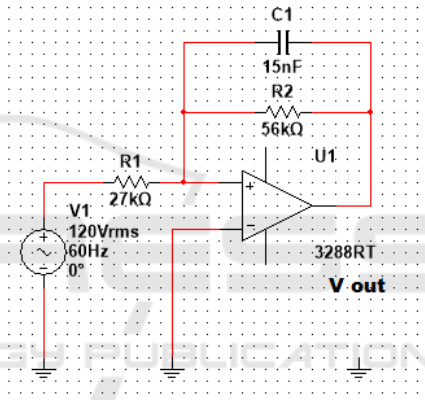

Figure 11: Low pass filter evolved circuit.

Figure 12 compares the frequency responses of the target and the evolved low pass filter indicating the good performance of the extrinsic platform.

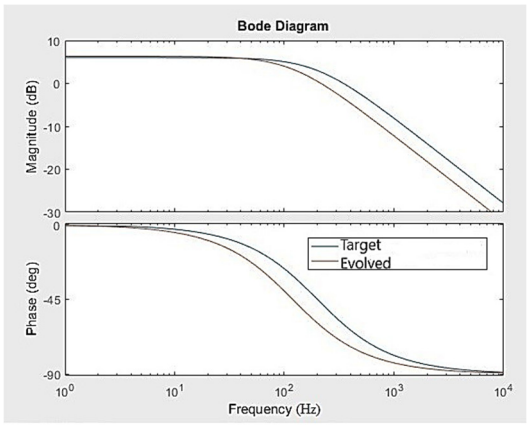

Figure 12: Low pass filter frequency response for target and evolved circuit.

Finally, figure 13 show the Bode diagram of the evolved low pass filter as conceived by the platform.

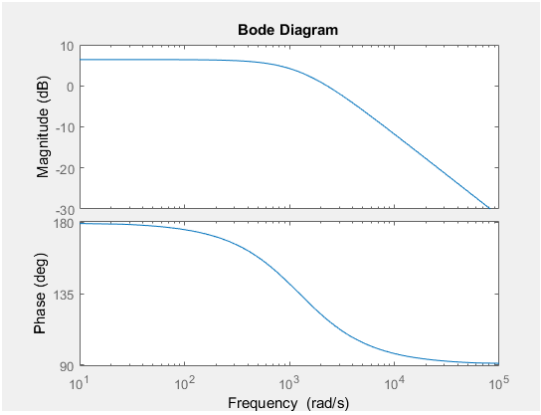

Figure 13: Low pass filter frequency response solution given by the platform.

4 CONCLUSIONS

The evolutionary process described in this study also has applications in the digital area, such as adaptive filters. The same process for the synthesis of the filters described is a viable tool for the design of other filters, such as Butterworth and Chebyshev filters. The same evolution procedure could be adopted with continuous genetic algorithms. This is not the case with the application of the GAOT package, which uses a binary genetic algorithm. It is important to point out that these algorithms can be modeled by a programmer in other programming languages, or a package from another language, such as R and Python. The most relevant fact of the procedure is the fit in the complete process, in which functions are used to generate considered curves and ideas and simulators for the observed curve, later establishing the comparison between the curves and assigning a representative value to the degree of similarity. In relation to continuous genetic algorithms, the concepts of elitism, mutation, selection and crossover have the same meaning in relation to those existing for binary algorithms. However, for a continuous algorithm, the parents are generated within a limit set by the code. The crossover generates heirs within the region by the parents and it is the mutation, in a manner analogous to its behavior described for binary genetic algorithms, that will produce solutions outside the limits established by the parents. The development steps follow the same method. In the case of adaptive filters, this technique has the potential to be applied to purely digital filters, without component synthesis. An adaptive filter is built on the basis of an FIR or IIR filter. MATLAB has work tools for both. There is, in the audio field, a growing market for the use of artificial intelligence applied to digital processing. MATLAB has many filter functions that can be used on this extrinsic platform. Finally, it is noted that this platform can be used for other applications in subsequent studies, and that MATLAB has all the necessary basis for each step of the described evolutionary process.

The developed project had as main objective the application in synthesis of electronic circuits used in the audio area. Through the design of an extrinsic platform, it was possible to obtain optimal solutions for known topologies, without considering the evolution of the topology. As a future work it is being considered by the authors to include the evolution of the topology as well. That will need a reformulation the optimization process due to the expansion of the search space. All this study was carried out using the MATLAB software and its SIMULINK extension, evolving only component values and comparing the behavior of the transfer functions obtained for the solution with a target behavior, generated by mathematical functions of MATLAB filters. For some circuits, prior knowledge of the transfer function was necessary for their design. However, it is emphasized here that this fact does not create difficulties for the user of the platform, since it is possible to obtain the transfer function of any circuit through the joint work tools of SIMULINK and SIMSCAPE.

Curves close to those idealized were obtained, with satisfactory efficiency, which demonstrates the ability of the developed platform for situations of reasonable engineering complexity. The combination of MATLAB tools with the evolutionary concepts of the genetic algorithm allowed to obtain good solutions for previously known circuits in relation to their topology. In fact, the only limitation of this process is that it starts from a known topology, without evolving it, which would increase the possibilities of solutions, but justified by the established configurations in the market. Finally, it was possible to verify the ability to search for optimal solutions through evolutionary electronics for the synthesis of analog circuits, and also that the proposed evolution technique has other fields of application.

ACKNOWLEDGEMENTS

This study was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001.

REFERENCES

Mirjalili, S. and Jin-Song, D., 2020. Multi-Objective Optimization using Artificial Intelligence Techniques. 1st. ed., Warsaw, Poland, Springer.

Coello Coello, C. A., 2013. Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Vol. 36 of the series Computational Methods in Applied Sciences pp 3-18.

Coello Coello, C. A., 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems, Volume 1, Issue 3, pp. 269–308.

Greenwood, G. W., Tyrrel, A. M., 2007. Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems". IEEE Press Series on Computational Intelligence. David B. Fogel Series Editor.

Reorda, M. S, Sterpone, L., Ullah, A., 2017. An Error-Detection and Self-Repairing Method for Dynamically and Partially Reconfigurable Systems. IEEE Transactions on Computers, Volume 66, No. 6.

Sedgewick, R., (1998). Algorithms, 3rd Edition, in C, Parts 1-4: Fundamentals, Data Structures, Sorting, and Searching. Reading, MA: Addison-Wesley.

Coelho P., M. do Amaral J. and Bentes M., 2021. Electronic Circuits Extrinsic Evolutionary Platform. In Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS, pp. 752-759.