Waveforms Synthesis by Evolutionary Processes

José Fornari [1] **, Jônatas Manzolli** [1,2] **, Adolfo Maia Jr.** [1,3]

[1] Núcleo Interdisciplinar de Comunicação Sonora (NICS), UNICAMP,
13.081-970, Campinas, SP, Brazil.

[2] Instituto de Artes, UNICAMP,
13.081-970, Campinas, SP, Brazil.

[3] Departamento de Matemática Aplicada, IMECC, UNICAMP,
13.081-970, Campinas, SP, Brazil.

`fornari,jonatas,adolfo@nics.unicamp.br`

Abstract. *We present a model for sound synthesis based on Genetic Algorithms (GAs) and the correspondent algorithm named ESSynth. It creates an evolutionary sequence of waveforms which converges to a Target Sound Set and can be used to several purposes such as real time sound environment, or even collect the resultant sounds for an off-time composition, electroacoustic music,etc.*

1. Introduction and Motivation

GAs have been, in the past few years, frequently applied to generate and manipulate evolutionary musical material. Biles [Biles, 1990] presented a genetic algorithm-based program that mimics a student learning to improvise jazz solos under the guidance of a human mentor. In Horowitz's [Horowitz, 1994] development, an interactive system uses GAs to develop a criteria to distinguish rhythmic patterns producing a large number of variations. One of the present authors has also studied applications of GAs to interactive composition [Manzolli et al., 1999]. Similar to the approaches described above, our previous research used MIDI data to control music events in real time. Yet, using a different heuristic, we created a system named Vox Populi [Moroni et al., 2000], a hybrid system formed of an instrument and a compositional environment. Evolutionary Algorithms were used by Johnson [Johnson, 1999] to develop a computer system for sound design. Roads [Roads, 1994] used genetic algorithms in granular synthesis to facilitate the regulation of its parameters.

Since it is, in general, difficult to combine quantitative and qualitative descriptions of a given sound, we apply the concept of Evolutionary Systems to develop a methodology for sound synthesis or, generally speaking, for Timbre Design. This method was named as ESSynth and is the principal point of this paper. ESSynth uses a set of *Target Waveforms* (in EA jargon, target population)to describe a timbral tendency generated out from an initial set of waveforms (initial population), new variants (generations) *similar* to those ones in the target. This similarity is measured by evaluations of a *Fitness Function*. ESSynth can be seen as a ruled-based algorithm that uses an implicit set of rules for generating waveform variations. In this way we want to formulate mathematical and computational models, in which we can define suitable genetic operations, such as crossover and mutation, operating as waveform transformations as well as defining measures for waveform similarities. By controlling the Target and Initial Populations, it is possible to create organized sound patterns or, at a higher level, to compose a piece of music. For the latter, an external device linked to the ESSynth such as a Wavetable engine is necessary in order to playback the musical sequences in real time.

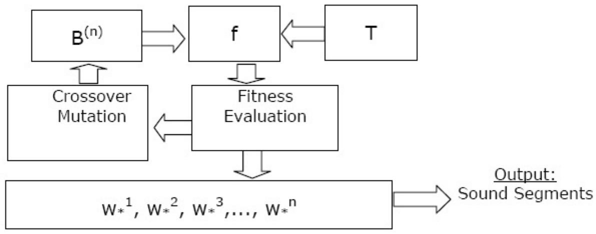

Figure 1: Basic ESSynth Diagram

In ESSynth the sound is dynamically generated by evolution, rather than being a static outcome. So GA are used as to generate sonic process instead of searching for an optimal sound. Some ESSynth characteristics are:

1. ESSynth has an intuitive controllability and easily generates ”rich” sounds. For example, it merges the best of linear and non-linear synthesis methods.
2. ESSynth is a non-monitored learning machine that automatically pursues the best sound in a population (i.e. Wavetable) and improves each waveform according to the user’s intuitive parameters, by handling the waveforms in the Target population.
3. The meaningful outcome is not only the sound segment but also the evolutionary trail of waveforms as time goes by. It comes along with the dynamic changing nature of sounds

2. ESSynth Description

There are three basic structures which comprises the kernel of ESSynth which we list below.

1. The population of the n-th generation is given by a set of m waveforms $\mathbf{P}^{(n)} = \{\mathbf{v}^{(n,1)}, \mathbf{v}^{(n,2)}, \ldots, \mathbf{v}^{(n,m)}\}$. The number m of waveforms could change from population to population, but in this work, for the sake of simplicity, we keep it fixed. The initial population is denoted by $\mathbf{P}^{(0)} = \{\mathbf{v}^{(0,1)}, \mathbf{v}^{(0,2)}, \ldots, \mathbf{v}^{(0,m)}\}$.
2. $\mathbf{T} = \{\mathbf{t}^{(1)}, \mathbf{t}^{(2))}, \ldots, \mathbf{t}^{(m^*)}\}$, denotes the Target population. Observe that the target can have a diferent number of elements of the populations in each generation, that is $m \neq m^*$.
3. A function *f* named *fitness function* used to evaluate the n-th generation’s best individual denoted by a vector $\mathbf{b}^{(n)}$.

A simplified diagram of ESSynth is shown in Fig. 1. Waveform variants are produced by applying the genetic operations crossover and mutation on the population of individuals, for each generation $\mathbf{P}^{(n)}$. So they are external interventions producing a modified population in order to be selected from it a new best fitted individual. An interesting ESSynth’s feature is to make waveform patterns dynamical sequences in real time. Biologic evolution produces species diversity. As a GA based application ESSynth creates and manipulates complex generations of sound material. In short, the operator crossover increases the waveform co-variance and mutation produces random population variations. These two genetic operations are defined in Φ, the collection of all finite sets (populations) of waveforms, as follows.

We show the algorithm that controls the selection of individuals from a generation to the next one presumely leading to the populations with more evoluted individuals which here means those ones with higher fitness values. The algorithm is as follows.

1. Generate an initial population $\mathbf{P}^{(0)}$ as well the Target population $\mathbf{T}$of waveforms. This could be done importing them from a wavetable or from another application as synthesizers software.
2. Define the Fitness Function.
3. Find the best element.
4. Apply the Genetic Operators crossover and mutation shape on the actual generation and obtaining the next population.
5. Evaluate the distance to the Target Population using the Hausdorff Distance.
6. Repeat the steps (3), (4) and (5) for each generation.
7. Halt the process when the distance is less or equal a prefixed number ϵ

3. Experimental results and analysis

In this section we present a set of experiments we produced to evaluate the potential of the method. These experiments can be found at the link http://www.nics.unicamp.br/ fornari/essynth1/.

We developed a methodology to evaluate and understand how the ESSynth works. This analytical process was based on the following criteria:

1. generate a family of sounds using populations in which the target and population have a very clear spectral discrepancy.
2. choose an individual in the population and plot his spectral evolution using a sonogram.
3. generate a sonogram representing the best individual spectral evolution.
4. compare (b) and (c) to evaluate similarities between the evolution of an individual and the overall evolution of the best one.

We run some experiments to evaluate the spectral evolution of the method using sine waves and white noise. The population of sine waves was generated constructing a scale using a logarithmic spread of frequencies within the interval $[fmin, fmax]$, where $fmin = 80Hz, fmax = 4000Hz$. Eq. (1) describes how we generated the initial population and Figure 2 shows the sonogram of a population generated by this formula.

$$P(i,j) = \sin\left[2\pi\left(\frac{N}{2M}\right)\left(\frac{f_{max}}{f_{min}}\right)^{\left(\frac{1}{M}\right)}\left(\frac{j}{f_s}\right)\right] \qquad (1)$$

where $i = 1, \ldots, M$, where is the number of individuals in the population as presented in Table 2.0 , $j = 1, \ldots, N$, where is the number of samples in the waveform as presented in Table 2.0.

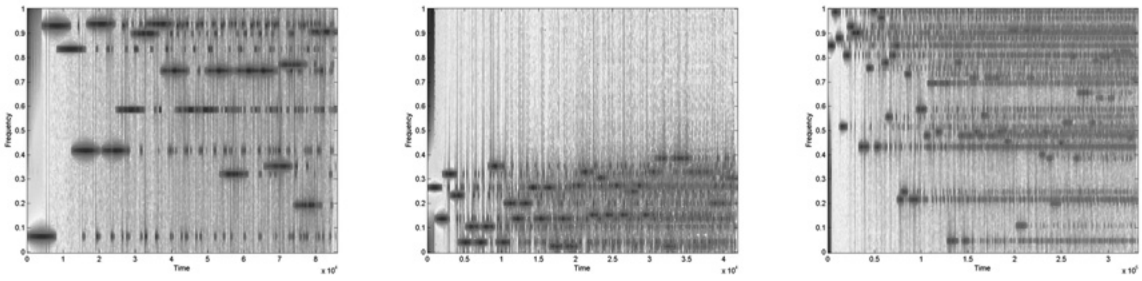

Figure 2: The spectral evolution sonogram of the best individual for three experiments.

4. Conclusion and Further Research

As far as our experimental results had shown, ESSynth is able to generate interesting sound results that dynamically change in time. The evolutionary characteristic of ESSynth methodology allows to explore time domain in a new fashion of sound synthesis, given by the continuous change on the sound segment. So its output can be used to real time composition as well as off time electroacoustic music. These perceptual sound changes are guided by the sound characteristics expressed by the individuals of the target set. If the target set remains unchanged the sonic evolution tends to converge, to a unique sound segment. Nevertheless, ESSynth allows the user to change the Target in real time driving the evolutionary sound to explore other sound regions. Also it is important to stress that we have used Euclidian Metric for its simplicity. Actually there is an infinite number of distinct metrics which could be used. The user/musician must choose which one is best for his/her purpose.

A possibility for further research is to have a population set that is not fixed in size. This implies in having a floating size population, what is closer to the biological reality. In addition we can think of a ripping period for individuals in the population, before they are able to reproduce. This may be also named as *childhood*. During this period the individual would be able to acquire information (learning) from other individuals within the population. Our experimental results shown here point that ESSynth could be a very interesting method for sound synthesis mainly for real time applications. Finally, it worth to mention that aesthetic issues related to sound outputs of ESSynth, although important to a concrete use of it, deserves a deeper investigation which is out of scope of this work.

5. Acknowledgments

This work was supported by São Paulo State Research Foundation (FAPESP) and CAPES (Brazil). A.M.J is gratefull to the Computer Music Team of Plymouth University for the hospitality during the making of this work.

References

Biles, J. (1990). A genetic algorithm for generating jazz solos. In *The Language of Electroacoustic Music*, pages 61–93. Ed. Emmerson.

Horowitz, D. (1994). Generating rhythms with genetic algorithms. In *Proceedings of the 1994 International Computer Music Conference*, pages 142–143.

Johnson, C. G. (1999). Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In Wiggins, G. A., editor, *Proceedings of the AISB Workshop on Articial Intelligence and Musical Creativity*, Edinburgh.

Manzolli, J., Moroni, A., Von Zuben, F., and Gudwin, R. (1999). An evolutionary approach applied to algorithmic composition. In *Proceedings of VI Brazilian Symposium on Computer Music*, pages 201–210, Rio de Janeiro.

Moroni, A., Manzolli, J., Von Zuben, F., and Gudwin, R. (2000). Vox populi: An interactive evolutionary system for algorithmic music composition. *Leonardo Music Journal*, 10:49–55.

Roads, C. (1994). Genetic algorithms as a method for granular synthesis regulation. In *Proceedings of the 1993 International Computer Music Conference*, Edinburgh.