Evolutionary Musique Concrète

Cristyn Magnus

CRCA and Department of Music, UCSD

Abstract. This paper describes a genetic algorithm that operates directly on time-domain waveforms to generate electronic music compositions. The form of these pieces is derived from the evolutionary process. Recorded sounds are treated as chromosomes. The sounds evolve in a world that consists of multiple locations. Each location has its own fitness function and mutation probabilities. These can change over the course of the piece, producing musical surprises. The aesthetic motivation of the work is discussed and the results of the algorithm are described.

1 Background

Although electronic music experiments had been going on since the development of the telephone, a great breakthrough came in 1948, when Pierre Schaeffer broadcast his early studies in *musique concrète* on Radio-diffusion-Télévision Française[1]. *Musique Concrete* is a genre in which composers manipulate recordings of actual sounds rather than notes. Composers who use notes deal with abstract symbols that represent large categories of possible sounds; performances are unique interpretations of the symbols. A composer of *musique concrète* produces a definitive recording that is the piece; at performances, the recording is simply played. Techniques for composing with actual sounds give composers access to an extremely wide array of timbres—anything that could be recorded or brought out of a recording through manipulation. We are no longer restricted to pitches and rhythms that can be written using traditional western notational symbols.

Since the incorporation of recorded sounds is pervasive in contemporary electronic music, it is ironic that little attention has been given to developing techniques for manipulating recordings with genetic algorithms. Most research applying genetic algorithms to music has focused on symbolic music (see [2] for a review). Some research has broached the issue of timbre exploration through synthesis, but direct manipulation of recorded sounds has not been addressed. Johnson[3] [4] and Dahlstedt[5] [6] use interactive genetic algorithms to explore synthesis parameters. Horner, Beauchamp, and Packard[7] derive novel sounds with an interactive genetic algorithm that applies filtering and time-warping operations to populations of synthesized sounds. This comes closer to addressing recorded sounds, since filtering and time-warping need not be applied exclusively to synthesized sounds. These researchers all work to produce novel sounds that can be worked into later compositions. For a series of recent compositions, I have developed a technique that would allow me to use genetic algorithms to produce a series of pieces constructed from found sounds whose form would be derived from the evolutionary process.

2 A Genetic Algorithm that Operates on Time-Domain Waveforms

Since conventional genetic algorithms are meant to be applied to discrete symbols, applying them to sounds requires some modification. In my description of these changes, I will try to distinguish between practical choices that can be transferred to other musical projects and aesthetic choices that result in the characteristic sound of my pieces.

2.1 Representation

In a typical genetic algorithm, parameters are mapped onto genes and the ordered collection of genes forms a chromosome. Usually all chromosomes in the population have the same number of genes. My technique operates directly on digitized waveforms that can have arbitrary lengths. Each chromosome is a time-domain waveform. Using instantaneous samples as genes would be a bad idea: sexual reproduction would introduce clicks; mutation would introduce noise. So in my algorithm, there is no analysis and there are no discrete genes. Instead, a hybrid approach to genes is adopted. For the purpose of calculating fitness, samples are treated as genes. For the purpose of sexual reproduction and mutation, segments of waveform bounded by zero crossings are treated as genes.

Typically, a genetic algorithm runs for many generations. The initial population and any intervening generations are discarded; a representative member of the final population is chosen as the product of the algorithm. My algorithm produces a piece of music whose formal structure is a product of the evolutionary process. Each waveform produced by the algorithm becomes part of the final piece. A piece begins with the simultaneous playback of the initial waveform population. Whenever a waveform finishes playing, a new waveform is generated to take its place. The instant before a waveform's playback begins, its fitness is measured. Each waveform's playback volume is weighted by its fitness.

Since the output of the algorithm is a piece of music, choices regarding output representation are primarily aesthetic. If I wanted a piece with a different formal structure or simply a tool to generate sonic material to use elsewhere, I would make different choices.

2.2 Fitness

The choice of fitness function is primarily aesthetic. The purpose of a fitness function in my algorithm is simply to provide directionality for pieces produced by the algorithm and to determine the volume at which each waveform is played back. It is important that some waveforms be fitter than others for natural selection to take place. The fitness function is based on the correlation between waveforms in the population and a specified target waveform. Formally, this can be written as

$$Fitness = \frac{waveform \cdot target}{\|waveform\| \; \|target\|} b^n \tag{1}$$

where n is the number of times the waveform has reproduced and b is a parameter between 0 and 1. For $b = 0$, a waveform will never reproduce twice. For $b = 1$, a waveform's fitness is not reduced by reproduction. The b^n modifier is to encourage biodiversity (see §4.2).

Although a stripped down version of the algorithm can, under appropriate circumstances, produce sounds that can be recognized as imitations of the target waveform, this is not the compositional goal. The population is never expected to converge to some target. The biodiversity modifier lowers fitness each time a waveform becomes a parent to prevent the offspring of a handful of extremely fit individuals from dominating the population. In addition, the compositional framework for the piece (§3) has high-level control over the fitness function, which can change over the course of the piece.

2.3 Reproduction

Sexual reproduction is carried out by splicing genetic material from two individuals to produce one individual in the next generation. For each offspring, two parents are selected from the population. The probability that an individual will be selected as a parent is based on its fitness. Each parent is divided at some randomly selected crossover point. The location of the crossover point is adjusted to make sure it falls on a zero crossing. The first part of one parent is spliced to the last part of the other parent (figure 1). Because the crossover point is randomly selected and can be different for each parent, offspring can be arbitrarily short or potentially as long as the combined lengths of both parents. I could have used a fixed crossover point, but I felt this was an opportunity to introduce rhythmic interest.

2.4 Mutation

Mutation occurs immediately after the offspring is produced, before its playback begins. Each segment of waveform between zero crossings has a slight probability of mutating. This mutated segment of waveform can include multiple zero crossings. Larger mutations are more perceptually relevant; that is, it is possible for a listener to identify mutated segments and sometimes even the type of mutation. Smaller mutations tend to denature the original sounds and produce waveforms that sound more like the target waveform.

A typical mutation function adds a random number to a gene. We can extend this concept to waveforms by adjusting a waveform's amplitude (figure 2a). This is done by selecting a random number and multiplying each sample of a waveform segment by that number. Another way of extending this concept is to raise each sample of a waveform segment by a power (figure 2b). To prevent

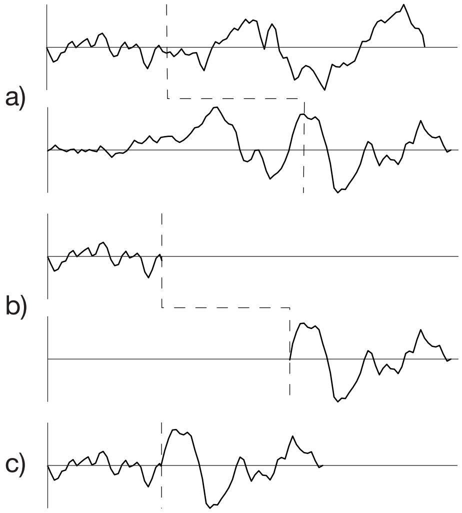

Fig. 1. a) Two parent waveforms (*solid line*) with their randomly selected crossover points (*dotted line*). b) The crossover point adjusted to fall on zero crossings. c) The child waveform

the exponentiation from severely amplifying or attenuating the segment being mutated, each segment is normalized after exponentiation so that it retains its original maximum amplitude.

We can think in terms of time rather than amplitude and resample a segment of waveform to lengthen it, making it lower in pitch, or to shorten it, raising its pitch (figure 2c).

Because mutation is applied to segments of waveform, rather than individual genes, we can draw inspiration from the types of errors that happen in actual gene transcription. Mutation functions can reverse a waveform segment (figure 2d), remove a waveform segment entirely (figure 2e), repeat a waveform segment a random number of times (figure 2f), or swap neighboring waveform segments (figure 2g).

3 Compositional Framework

In a single, unchanging environment, the algorithm described above would eventually converge to a local minimum where all individuals would have roughly the same length as the target waveform and would have acquired some of its amplitude envelope and frequency characteristics. To create formal compositional structure, I define a *world* in which the waveforms evolve. A world consists of multiple distinct environments that change over time.

For a given piece, the world will be characterized by some number of locations. These locations may be mapped spatially onto speakers. The *environment* at each location will initially be defined by some target waveform and some set of

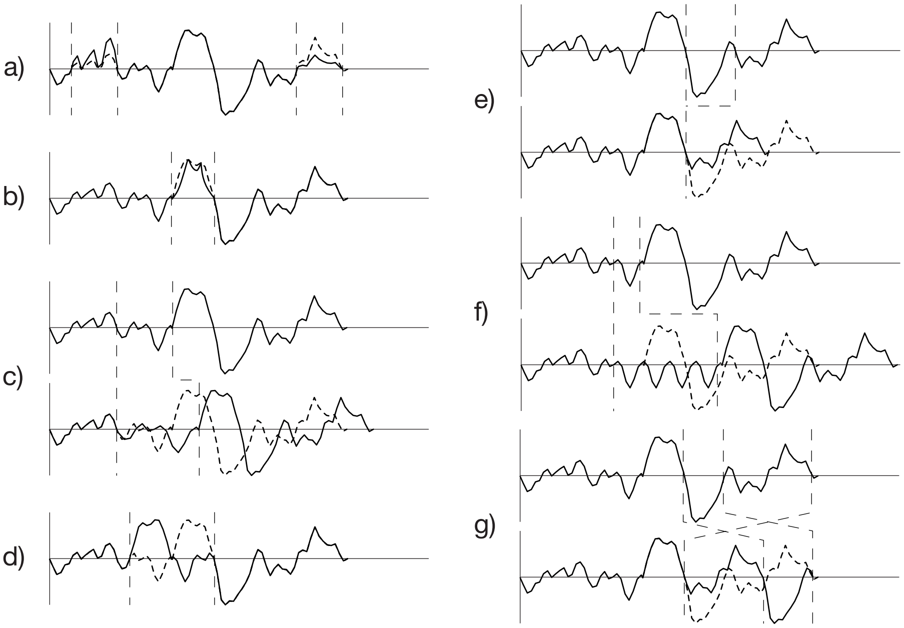

Fig. 2. Mutation operations showing the original waveform (*short dashes*) and the resultant waveform (*solid line*) with the mutation boundaries (*long dashes*): a) amplify b) exponentiate c) resample d) reverse e) remove f) repeat g) swap

mutation probabilities. Immediately after an individual is created, it has a slight chance of moving to another location. If it migrates, it will pan from one speaker to the other over the course of its playback. It will be considered to be in the second location for its entire duration and will have its fitness determined there. It will be given the opportunity to reproduce in the second location, but not the first. In this way, sounds with new characteristics will enter each location, enhancing biodiversity.

The world will be characterized by probabilities of change. Both target waveform and mutation probabilities can change whenever a new waveform is created. There are two sorts of changes that environments can undergo. One is the slow drift that is seen in ice ages: these take place over an enormous amount of time from the perspective of individuals but happen many times over the evolution of a species. This is simulated by slowly cross-fading between two target waveforms. The other is the drastic change that results from catastrophic events, such as fire decimating a forest, causing it to be replaced by grassland. This is achieved by replacing the target waveform with a completely different waveform.

The changing environment prevents the population from strongly resembling the target waveform. The goal is to present the process, not draw attention to the underlying environment. Catastrophic environmental changes lead to musical surprises that reveal subsets of the population that were previously too unfit to be heard above the dominant sounds. Migration can have similar effects; it also increases biodiversity, which means there are always sounds in each location that can take advantage of the changing environment.

4 Results

4.1 General Description of Output

As evolution occurs, all of the waveforms in the population are written to a single sound file with each individual waveform weighted by its fitness. This weighting causes fit individuals to rise to prominence. Each time a waveform ends, a new individual is generated from the population. The new individual's playback begins immediately at the end of the waveform it replaces. Because the initial biodiversity is very high, the beginning of the output file is a wash of textures reminiscent of the timbres of the initial population. Within a few generations, a few fit individuals dominate the mix, causing a sound in which particular features of the initial population can be identified.

As evolution progresses, qualities of the initial population are preserved but are increasingly transformed through reproduction and mutation as the population takes on properties of the target waveform. The similarity to the target waveform depends on the type of mutation used, on the probability of mutation, and on the amount of time over which evolution occurs.

4.2 Biodiversity

In order for a piece to be musically interesting, biodiversity must be maintained. Since output is weighted by fitness, only fit sounds are heard distinctly. The truly musical moments occur when previously unfit sounds become fit, either through a changing environment or migration. Novel sounds bloom out of the sea of sounds and affect what is heard after they become fit.

4.3 Effects of Mutation on Output

Each type of mutation has a characteristic sound that can be readily heard if a population evolves with only that type of mutation. Amplification changes the population in two ways. The amplitude envelopes of individuals in the population tend towards the amplitude envelope of the target environment. Portions of individuals that are in phase with the target will be amplified, while portions that are out of phase will be attenuated. Exponentiation is very similar to amplification in its behavior, but it is much more invasive; it significantly alters the timbre of the waveform. Resampling allows pitch to become closer to the pitch of the target waveform.

The quality of the biologically inspired mutations (reverse, remove, repeat, swap) depends largely on the number of neighboring genes grouped for mutation. Application to large segments of the waveform leaves the waveform more recognizable but is less likely to add significantly to the fitness of the population. Given a population of individuals that are several seconds long, typically one or two lengthy mutations will audibly propagate to future generations over the course of a several minute piece. Application to very small segments of a waveform typically makes the original sound unrecognizable but is more likely to have a positive effect on fitness and be incorporated in the population.

When the biologically inspired mutations are applied to perceptibly large segments of a waveform, the function itself can be clearly identified. That is, the a listener can tell that a segment of a waveform has been reversed, removed, swapped with another segment, or repeated. When the grain size is fairly small, portions of the waveform tend to get shuffled around to more closely resemble the target waveform. Portions of a waveform that have been reversed tend to retain some quality that tells the listener that reversal has taken place, but the only biologically inspired mutation that has a significant fingerprint when applied to small segments of a waveform is repetition. Repetition creates pitch out of noisy segments of a waveform. When the grain size is small and the probability of mutation is high, repetition is effective at getting the population to denature to the point where the target environment can be recognized. For example, a listener unfamiliar with the target environment can identify the environment as a bell when listening to the evolution of a population of waveforms evolving with a bell as the target environment.[1]

4.4 Achieving Musical Results

Because the goal here is to make interesting music, rather than to attain a duplicate of some target sound file, I usually choose fairly small mutation probabilities and to apply mutations to fairly large segments of waveforms. This

---

[1] See `http://cmagnus.com/cmagnus/ga_results.shtml` for sample output.

allows the sounds to be quite recognizable, even several minutes into the output file. The migration of individual waveforms from one environment to another and the ability of environments to change over time significantly contributes to the musicality of the output. I chose probabilities for both migration and environmental change that caused the trajectory of the piece to change every few minutes. This prevented the population from being dominated by the offspring of a few individuals and becoming monotonous.

5 Conclusion

I have used this algorithm to produce several pieces and an installation that have been performed and well received.[2] Many listeners have expressed surprise that the pieces were algorithmically generated with no composer intervention beyond setting initial conditions. This speaks to the algorithm's efficacy in producing novel and pleasing musical results. Depending on the source sounds and initial probabilities that I choose, I can generate very different pieces that share the characteristic sound of the algorithm. Over the course of a typical piece, sounds from the initial population slowly evolve. Rhythms change gradually; different sounds from the initial population rise to prominence at different points; and the piece has clear directionality, punctuated by occasional musical surprises.

References

1. Griffiths, P.: A Guide to Electronic Music. Thames and Hudson (1979)
2. Burton, A.R., Vladimirova, T.: Generation of musical sequences with genetic technique. Computer Music Journal **23**(4) (1999) 59–73
3. Johnson, C.G.: Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In Wiggins, G.A., ed.: Proceedings of the AISB Workshop on Articial Intelligence and Musical Creativity, Edinburgh (1999)
4. Johnson, C.G.: Exploring sound-space with interactive genetic algorithms. Leonardo **36**(1) (2003) 51–54
5. Dahlstedt, P.: Creating and exploring hute parameter spaces: Interactive evolution as a tool for sound composition. Proceedings of the International Computer Music Conference (2001) 235–242
6. Berry, R., Dahlstedt, P.: Artificial life: Why should musicians bother? Contemporary Music Review **22**(3) (2003) 57–67
7. Horner, A., Beachamp, J., Packard, N.: Timbre breeding. Proceedings of the International Computer Music Conference (1993) 396–398

---

[2] See `http://cmagnus.com/cmagnus/comp/gasketch.shtml` for a short piece.