# Generating Rhythms with Genetic Algorithms

Damon Horowitz
MIT Media Laboratory, 20 Ames St. E15-488, Cambridge, MA 02139
damon@media.mit.edu

## Abstract

Interactive genetic algorithms (IGAs [Smith 91]) are explored as a mechanism for controlling music generation systems. IGAs are well-suited for this task because they allow a user to simply choose which musical material he likes, without necessarily understanding the details or parameters of how the system is generating this material. Specifically, my system uses an IGA to learn a user's criteria for generating percussion textures. As the system learns (develops an increasingly accurate model of the function which represents the user's criteria), the quality of the rhythms it produces improves to suit the user's taste. This paper describes two version of the system: 1) the initial version which demonstrates the efficacy of the GA to produce near-optimal desired rhythmic figures in a simplified rhythmic domain; 2) the current version which is designed to be of practical use in evolving complex rhythmic textures. This approach is largely inspired by the work of Richard Dawkins, who succinctly summarizes the attraction of IGAs for artistic endeavors: "Effective searching procedures become, when the search space is sufficiently large, indistinguishable from true creativity" [Dawkins 86].

## Introduction

This project is motivated by the fact that complex music generation systems (such as [Rigopolous 1994]), which rely on many input parameters in order to determine their output, require a simple and intuitive mechanism for providing a user with control over the system's behavior. IGAs provide the user with a fundamental mode of control, namely that of indicating a subjective evaluation of the system's output, and thus would help to make such systems useful compositional aids. The long term goal for this project is therefore to use the IGA as one element of an interface to such a system; the IGA would serve as a module which could guide the generation of material similar to that for which the user indicates a preference. To date, the project has focused on the smaller goal of using an IGA to control the generation of unpitched rhythmic fragments, proceeding in two distinct stages.

The initial work on this system was geared towards determining whether an IGA would be successful at producing desired rhythmic fragments when using an extremely simple literal representation of rhythms. This is a difficult learning problem because of the subjectivity and variability of a user's criteria when evaluating rhythms; however, I found that by using several modes of interaction (described below) the IGA was able to successfully produce goal rhythms. Once this theoretical ground was established, the second stage of the project entailed the development of a practical application for controlling the generation of complex percussion textures by using the IGA with a simple parametric rhythm generation system.

## Initial Version

Initially, rhythms were represented as one measure long sequences of notes and rests occurring on pulse subdivisions of a beat; I only dealt with a specific subset of the enormous class of rhythms, in order to provide a well-defined domain for the application and testing of the learning algorithm (the GA). The basic mode of interaction is for the user to playback each of the rhythms in a randomly generated population, and then subjectively assign them fitness values based upon their satisfaction of his criteria. The system then uses standard GA selection (with fitness scaling), reproduction (with crossover monitors), and mutation operators to produce the next generation of rhythms (see [Goldberg, 1989] for a good introduction to GAs).

There are also several objective fitness functions with which the system can automatically evolve generations of rhythms: syncopation, density, downbeat, beat repetition, cross rhythm, and cluster functions are currently included. Each of these functions represents an axis in a feature space which is useful for distinguishing rhythms. (While these are only a few of the many possible objective functions that could be implemented, they provide a rich set of possibilities with which to begin exploring.) The user can specify the ideal target value for each of these fitness functions, and also their relative importance (weighting of coefficients) in determining the overall fitness of a rhythm. The system then automatically evolves the indicated number of successive generations, using the objective fitness values to determine selection.

The system also makes use of a meta-level genetic algorithm designed to evolve populations of parameters (target values and weights) to the objective fitness functions defined above. This is motivated by the research done in the application of genetic algorithms to the k-nearest-neighbor technique of classification [Punch et al. 93]; each meta-level individual represents a warping of K-NN space, such that the fitness of each individual is determined by how well its warping of the feature-space helps to discriminate useful features, and thus correctly perform classifications. Evolving populations of meta-individuals allows a user to quickly reduce the search space without having to directly specify values for the objective functions; the user evaluates the families of rhythms evolved by each of the meta-individuals.

This combination of methods proves to be a powerful hybrid approach, one which allows for greater coverage of the search space than would have been possible ordinarily using a small population (which is necessitated by most IGAs, and is particularly important when dealing with sequential acoustic data), and more efficient convergence on a satisficing solution. The system is able to successfully converge on near-optimal solutions (that is, to find goal rhythms which satisfy the test users' criteria) after about fifty user-evaluations of rhythms.

### Practical Application

In order to produce complicated multi-timbral percussion textures, I created a simple parametric model of a rhythm generator. The generator takes as input parameter values defined over the time span of one measure, and outputs a measure long rhythm. The generator is designed to produce fragments of common simple rhythms, such as were evolved successfully by the initial system; therefore, this later system can be seen as a continuation of the earlier system, using as starting materials those fragments which had to be evolved in the earlier system. For example, the generator can produce streams of sixteenth notes with an accentuation pattern that varies according to the specified degree of syncopation; such an example of a simple rhythmic figure would have been evolved in the earlier system, but can now be instantly produced when the parameters are set correctly.

The new genotypes for the rhythms consist of values for the amount of rhythmic "activity", syncopation, and accent structure over the course of a measure. The activity is on a scale which varies continuously from triplet sixteenth notes to quarter notes, with intermediary values which use accent structures to give the effect of being "between", say, sixteenth- and eighth- note pulse levels. The syncopation determines the degree of accentuation (or deaccentuation) of sequential notes with respect to the beat. The accent structure determines both the depth of the accents applied by the syncopation and activity parameters, as well as a separate "signature" pattern of accentuated "hits" in the measure. While this appears to be a drastically oversimplified representation of rhythms, the system demonstrates that the variation of these parameters over a repeating period can produce the complete range of possible quantized triple or duple rhythmic figures.

The application currently developed with this representation defines each individual in the population as a collection of "instruments", each of which is assigned its own percussion timbre and set of parameters (defined over a measure). Thus, each individual represents an ensemble of percussion instruments, and produces a distinct rhythmic texture when generated. To produce a new generation of textures, the individuals with the highest fitness from the previous generation are reproduced. The crossover operation takes place within similar drum groups (for example, a hi-hat pattern in one individual may be crossed with the ride cymbal pattern in another). By restricting crossover to common drum groups, the similarity of the offspring textures is increased; the effect is that the new textures tend to have some of the same instrument "roles" (drum groups) as the parents, and the patterns for these roles are reasonable modifications of the originals (formed by combining the input of both parents).

The user interacts with the system following the model presented in the initial version: the user can listen to each of the individuals in the population (in real-time; the rhythmic textures can be varied instantaneously), indicate his preference for them, and then breed another generation. Repeated iterations of this cycle by several test users have produced a wide variety of percussion textures; while it is difficult to objectively evaluate their quality, I can say with certainty that they rival the carefully prepared demo sequences distributed with most drum machines! More importantly, the evolved rhythmic textures satisfy the user's criteria by continually improving to suit his indicated preferences for the prior generation. Current work includes developing automatic fitness functions appropriate for the multi-timbral version. While the GA itself is mechanically quite simple, it is important to note that the implementation of appropriate fitness functions is difficult, and largely determines the musicality of the output.

### Conclusion

Music is an excellent example of a domain in which it is easy for humans to state their subjective opinion about a specific example, but difficult to say exactly what it is about the music that makes it satisfactory -- such a domain is exactly the type of domain in which an IGA is most useful. This project has demonstrated that IGAs are a powerful tool for controlling parametric music generation systems, enabling users to quickly create a variety of starting materials (or variations on existing materials) which can then be precisely tuned using traditional sequencer editing tools. The next application to develop will apply the current approach to a larger-scale music generation system (including tonal considerations for bass-lines, etc.). The major future improvement of the IGA will involve adding the capacity for the system to learn to design its own fitness functions to represent features characteristic of rhythms selected by users in past sessions by identifying patterns in their indicated preferences.

### References

Dawkins, R. 1986. “Accumulating Small Change,” in The Blind Watchmaker. New York: W.W. Norton and Co.

Goldberg, D. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. University of Alabama, Addison Wesley.

Punch, W. et al 1993. Further Research on Feature Selection and Classification Using Genetic Algorithms. Fifth International Conference on Genetic Algorithms.

Rigopolous, A. 1994. "Parametric Control of Seed-Generated Music". Masters Thesis, MIT Media Lab.

Smith, J. 1991. Designing Biomorphs with an Interactive Genetic Algorithm. Fourth International Conference on Genetic Algorithms.