SELECTION SONGS: EVOLUTIONARY MUSIC COMPUTATION

by Lee Spector, Jon Klein, and Kyle Harrington

In this article we provide a brief introduction to the use of evolutionary computation in the arts, focusing on two approaches that we have taken in our own work on the evolution of music-making systems.

Evolutionary Computation

Evolutionary computation is an area of computer science in which methods inspired by genetics and evolutionary biology are harnessed to explore vast search spaces and to solve computational problems. Several people have used these methods for artistic purposes, driven in part by the obvious creative track record of biological evolution.

One motivation for working with evolutionary models in the arts stems from our inability to build creative computer systems in many other ways—we are largely ignorant of the algorithmic principles of creativity, but evolution is crafty and evolutionary computation systems have a demonstrated capacity to find solutions beyond human devising. The hope, and in some cases the experience, is that systems based on evolutionary principles will produce artworks beyond the imaginative limits of the humans who design them. For surveys of some of this work see (Todd and Latham, 1992), (Bentley and Corne, 2001), (Whitelaw, 2004), and, for music applications in particular, (Todd and Werner, 1998).

Genetic programming is a subfield of evolutionary computation in which the evolving digital "organisms" consist of executable program code. In other words, it is software that evolves in a genetic programming system. If one wants to evolve music-making or art-making programs, rather than individual pieces of music or works of art, then it is natural to use genetic programming rather than other forms of evolutionary computation.

The genetic programming process begins with a population of randomly generated programs, made from the elements of an appropriate problem-specific "primordial ooze." Each program is then tested for "fitness," typically by running the program on appropriate problem-specific inputs, and the fitter programs are used to produce the next generation of programs. "Child" programs are produced from fit parents by direct copying ("survival of the fittest"), mutation (in which minor random changes are introduced), and sexual crossover (in which random components are combined from two fit parents). The process is iterated for many generations, until a sufficiently fit program is found; for details see (Koza, 1992).

Evolving Music

Several people have applied genetic programming to music; for example a search for "music" in the on-line genetic programming bibliography,[1] currently finds 18 matches. The relatively large number of applications to music may be due to the same factors that facilitated musical applications in the earliest days of computing technology, particularly the ease of representing musical scores using modest quantities of numerical data. But musical applications also present a unique challenge with respect to fitness assessment.

What makes a collection of artworks or an artist fit? In many systems the fitness of evolving art-making programs is assessed by human artists/users who observe and judge the quality of the artistic outputs that the programs produce. For visual art systems this can be done relatively efficiently by displaying an array of visual representations, sometimes nine or sixteen or even more, on screen at one time for rapid scanning and assessment. Typical applications of genetic programming use population sizes in the thousands, so human fitness assessment is a bottleneck even with this "parallelism" of display and judgment. Nonetheless, parallelism can sometimes help significantly.

Music presents a more severe problem, however, since our ears cannot make simultaneous sense of multiple independent musical pieces. Human fitness assessment therefore requires serial processing, which can be tedious. The genetic operators of mutation and crossover are "blind" to quality and often produce garbage, which is usually acceptable as long as they also occasionally produce improvements.

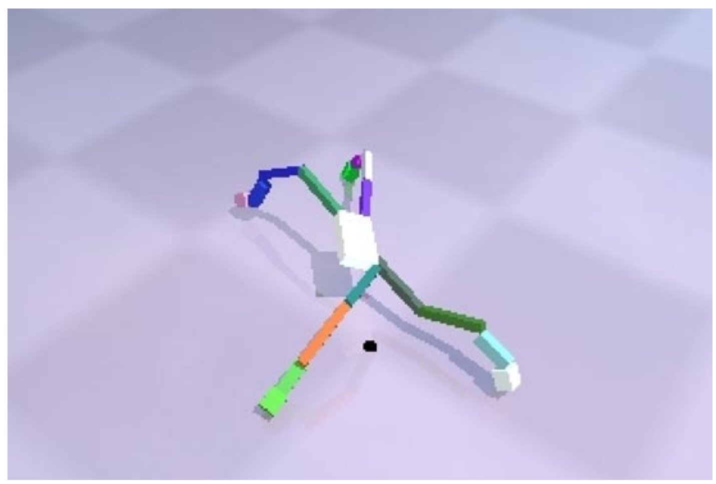

Figure 1. Evolved music-producing SuperDuperWalkers.

But serial human sorting of all of the garbage can be a major chore, and the long-term exposure to such garbage can numb one's ears to the sought-after quality so that it becomes progressively harder to make good discriminations. We have recently begun to explore the use of portable mp3 players to assist in this process, but this is not a complete solution.

Another approach to the fitness bottleneck is to automate fitness testing. But is it possible to write a program that makes aesthetic judgments? And would the judgments be good enough to drive the evolution of good art-making programs? Would it be as hard to make good judgments as it would be to produce good art in the first place?

Many of us in the field have been heartened by comments of arts

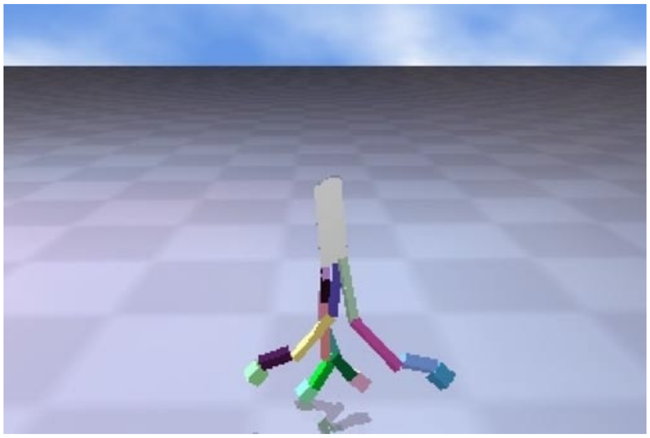

Figure 2. Evolved music-producing SuperDuperWalkers.

educators that aesthetic judgment is at least easier to teach than aesthetic production. For example Irwin R. Blacker, in *The Elements of Screenwriting*, writes, "The teacher can't give students an ear for dialogue, but he can show the differences between good and bad dialogue. He can't teach students how to invent a plot, but he can teach them to see the flaws and weaknesses of a plot" (Blacker, 1986).

Genbebop

Taking the possibility of cyber-criticism as a hypothesis we turned to jazz method books, primarily Baker's (Baker, 1988), for critical principles that might be built into a jazz cyber-critic. In particular, we attempted to extract unambiguous, algorithmic criteria for the assessment of musical responses to musical "calls." That is, we wanted rules to determine the quality of a musician "trading four," to use the common jazz idiom. The criteria that we developed were:

• Tonal novelty balance: A good response should contain a roughly even mix of interval sequences that can be found in the jazz corpus (for which we used a small collection of Charlie Parker tunes) and new interval sequences.
• Rhythmic novelty balance: The same as tonal novelty balance, but for rhythm.
• Tonal response balance: A similar balance should exist between material "echoed" from the call (in a call/response pair) and new material.
• Skip balance: Good melodic lines generally mix diatonic movement with larger intervals.
• Rhythmic coherence: Rhythmic sequences violating playability constraints should be penalized.

Using a fitness function based on these criteria, and a primordial ooze of music-making functions that would allow the evolved programs to transform musical calls into musical responses in a wide variety of ways, we were able to use genetic programming to evolve "fit" jazz-making programs (Spector and Alpern, 1994; sound samples available online[2]). To say that they were fit is not, however, to say that they were interesting! Indeed, evolution found "cheap" ways to meet the fitness criteria, for example by echoing half of a call and resting for the remainder. The cyber-critic was too easy to please and provided little more than a sanity check.

To extend this work we turned our attention to the production of smarter critics. We considered a variety of techniques, including the recursive application of evolution, using genetic programming to evolve critics. We settled instead on the use of neural networks that were trained to distinguish "good" from "bad" call-response pairs. The result was a critic that did appear, on informal testing, to have some sense. For example it appreciated some Charlie Parker sequences that it hadn't heard during training (along with at least one Jimi Hendrix solo), while it gave low scores to most random sequences. But the trained critic did not provide sanity checks, and musicians evolved using the critic as a fitness function produced unplayable rhythms over unplayable ranges. We solved this problem by using a "committee" of critics: we ran both our original rule-based critic and the trained neural network critic and averaged their judgments. This allowed evolution to produce a musician that was neither trivial nor psychotic, although our analysis was only preliminary (Spector and Alpern, 1995).

Selection Songs

More recently we have begun to explore more "ecological" notions of evolutionary music computation. In the work described above, the evolving programs never interact with one another or with the world (aside from a musical database); they simply respond to musical "problems" with musical "answers." In contrast, in some of our more recent work the evolving programs control complex virtual organisms that move around and interact in complex 3D virtual worlds such as "SwarmEvolve" (Spector and Klein, 2002; text, sounds and movies available online[3]). Music production is just one among the many activities of the virtual organisms in SwarmEvolve, along with flying, eating, reproducing, and sharing energy. In this system the music is "epiphenomenal"—it is a side-effect of the actions and interactions of the organisms. The music reflects the dynamics of the virtual world, but it has no effect on that world. (For other work on music-producing swarms see Blackwell and Bentley, 2002.) This work could be extended by allowing the evolving organisms to hear and to respond to the music. In this way the music might become an essential component of the virtual ecology.

In our most recent work we have continued to explore "epiphenomenal" music generated by evolutionary processes operating in 3D virtual worlds. We began with a system called "SuperDuperWalker" that was designed to support experiments on the evolution of locomotion (Spector et al., 2005).[4] In this system a genetic algorithm is used to search the space of possible limb numbers, proportions, and controllers for multi-legged "walking"

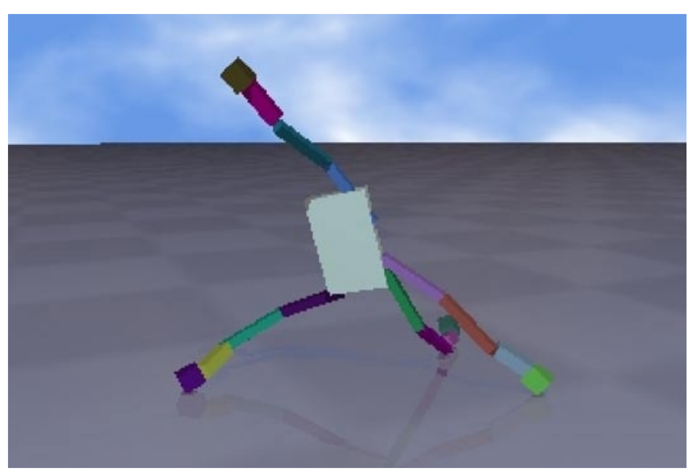

Figure 3. Evolved music-producing SuperDuperWalkers.

creatures, the fitness values of which are determined by distance traveled. The term "walking" is a bit misleading here—evolution is creative and often generates creatures that wriggle, roll, or even fling themselves across the virtual world.

We produced a musical version of SuperDuperWalker by associating sounds with all of the significant events in the virtual world. For example, when legs strike the ground they make piano tones, with higher tones resulting from strikes further from the creature's starting location. A continuous tone indicates the distance of the creature's body from the starting location, and additional sounds are generated by joint motions and by contact between the creature's body and the ground. These creatures are tested one at a time, with an additional sound indicating the transition between fitness tests. The character of the "songs" produced by the natural selection process change over evolutionary time, and the overall composition created by the evolutionary process develops in interesting ways. The resulting music is often interesting because it results from a complex, adaptive process that is constrained—as is human musical performance—by the physical properties of articulated 3D bodies. A sample movie, with sound, is available online,[5] and screen snaps are shown in Figure 1.

An extension of this work might use fitness functions that incorporate explicit assessment of the aesthetic value of the creatures' musical output, thereby increasing the selection pressure for the production of interesting music. The required assessments might be produced, for example, by the critics from the Genbebop project described above. This would introduce aspects of human musical idiom into the evolving virtual ecology.

Prospects

We have provided only a brief sampling of our projects in evolutionary music computation, which in turn constitute only a small sample of the work in this field that is being produced by a large and growing community of artists and technologists. Many other strategies exist for applying evolutionary computation technology to music and to other arts, and many of these strategies can be hybridized with one another in a variety of ways.

The prospects for this kind of work in the future are, in our view, very good. Evolutionary computation is a powerful technology with unusual creative potential. Applications of this technology to music and other arts are often relatively easy to engineer, as one can build on the long history of work on algorithmic music composition and use the many tools for digital audio and multimedia production that are now readily available.

We have found evolutionary computation to be an engaging and rewarding tool for artistic exploration, and we expect to see it used by more artists and musicians in the future.

Acknowledgments

The Genbebop project was a collaboration with Adam Alpern. We thank Rebecca Neimark for encouragement and suggestions. This material is based upon work supported by the United States National Science Foundation under Grant No. 0308540 and Grant No. 0216344, and by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30502-00-2-0611. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Baker, D., 1988. Jazz Improvisation: A Comprehensive Method for All Musicians. Revised Edition. Alfred Publishing Co., Inc.

Blacker, I. R. 1986. The Elements of Screenwriting. Macmillan.

Blackwell T.M. and P.J. Bentley. 2002. Improvised Music with Swarms. In Proceedings of the Congress on Evolutionary Computation, pp. 1462-1467. IEEE Press.

Bentley. P. J., and D. W. Corne, editors. 2001. Creative Evolutionary Systems. Morgan Kaufmann Publishers.

Koza, J. R. 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press.

Spector, L., and A. Alpern. 1994. Criticism, Culture, and the Automatic Generation of Artworks. In Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, pp. 3-8. AAAI Press/ The MIT Press.

Spector, L., and A. Alpern. 1995. Induction and Recapitulation of Deep Musical Structure. In Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music. pp. 41-48. AAAI.

Spector, L., and J. Klein. 2002. Complex Adaptive Music Systems in the BREVE Simulation Environment. In Bilotta et al. (eds), Workshop Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems, pp. 17-23. Sydney, Australia: University of New South Wales.

Spector, L., J. Klein, K. Harrington, and R. Coppinger. 2005. Teaching the Evolution of Behavior with SuperDuperWalker. To appear in Proceedings of the 12th International Conference on Artificial Intelligence in Education.

Todd, P. M., and G. M. Werner. 1998. Frankensteinian Methods for Evolutionary Music Composition. In Griffith, N., and Todd, P. M., editors, Musical Networks: Parallel Distributed Perception and Performance. The MIT Press.

Todd, S., and W. Latham. 1992. Evolutionary Art and Computers. London: Academic Press.

Whitelaw, M. 2004. Metacreation: Art and Artificial Life. The MIT Press.

(Footnotes)

[1] http://liinwww.ira.uka.de/bibliography/Ai/genetic.programming.html
[2] http://hampshire.edu/lspector/genbebop.html
[3] http://hampshire.edu/lspector/alife8-music.html
[4] http://hampshire.edu/lspector/superduperwalker.html
[5] http://hampshire.edu/lspector/selection-songs.html