GenJam in Perspective: A Tentative Taxonomy for Genetic Algorithm Music and Art Systems

John A. Biles

Department of Information Technology
Rochester Institute of Technology
Rochester, NY 14623
jab@it.rit.edu

Abstract

GenJam is an interactive genetic algorithm that models a jazz improviser and performs regularly in the author's Virtual Quintet. GenJam learns to improvise full-chorus solos under the guidance of a human mentor, and it trades fours in real time with a human performer in "chase" choruses. This paper first briefly describes GenJam's architecture, representations, genetic operators and performance characteristics, and then places GenJam in the context of a proposed taxonomy for GA-based music and art systems.

1 GENJAM REVEALED

Figure 1 summarizes GenJam's system architecture by showing the files it accesses and the humans with whom it interacts when playing a tune. Five of the files in Figure 1 (all but the two populations) provide information on the specific tune being played. The Rhythm Sequence is a standard MIDI file containing the rhythm section's accompaniment for the tune. The Head Sequence is a MIDI file containing GenJam's harmony part for the written melody or "head" of the tune. These two canned sequences are played during a tune while GenJam and the human performer improvise in real time.

The Chord Progression file provides the chord changes for the tune, as well as the tempo, rhythmic style (swing or even eighth notes), and pitch range GenJam should use. The Choruses file tells GenJam what it should do for each chorus (repetition) of the form defined by the chord progression. For example, a typical five-chorus tune might have GenJam play the head in the first chorus, rest while the human improvises in the second chorus, improvise a solo for the third chorus, trade fours with the human in the fourth chorus, and play the head again for the last chorus. The MIDI Parameters file configures a tone generator by specifying instruments, loudness, location in the stereo field, and as many as 30 other parameters for GenJam and each member of the rhythm section.

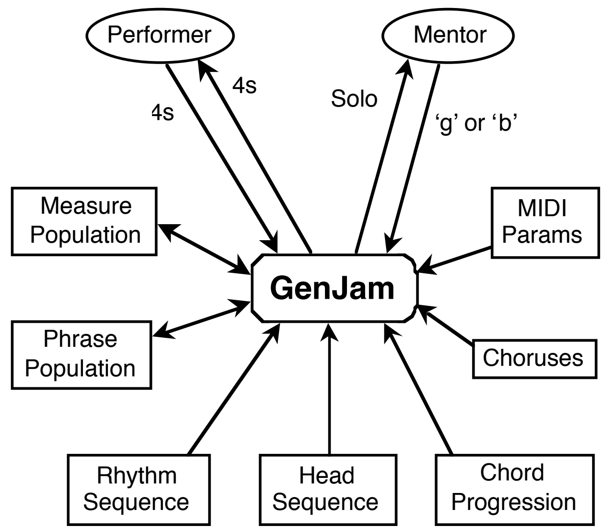

Figure 1. GenJam system architecture

The two population files represent a GenJam soloist and are independent of any particular tune. The Measure Population consists of 64 individuals, each of which decodes to a measure of eighth-note length events. Each event is represented by four bits, which results in 16 possible events and a chromosome length of 32 bits in 4/4 time. The 16 possible events are a rest event (coded as 0), a hold event (15), and 14 new-note events (1-14). A rest event is performed by sending a MIDI note-off message. A hold event is performed by doing nothing, which holds the previous event through the hold event's time window. A new-note event is performed by sending a MIDI note-off message, followed by a MIDI note-on message. The pitch for the note-on is determined by using the new-note event as an index into an array of 14 pitches constructed from the scale suggested by the chord being played at that moment in the tune. GenJam currently recognizes 17 different chord families [Biles 98]. This representation scheme is very efficient in that it integrates pitch and rhythmic structures in the same representation, and it is highly robust in that any sequence of 32 bits will decode to a measure of notes that are harmonically appropriate.

In other words, GenJam can't play a theoretically wrong note.

The Phrase population consists of 48 individuals, each of which decodes to a sequence of four pointers to individuals in the measure population. Since the measure population size is 64, six bits are needed for each pointer, which results in a chromosome length of 24 bits for phrase individuals. A full-chorus solo is constructed by randomly selecting enough phrases to cover the form defined in the chord progression. For example, a chorus of a typical 32-bar popular song form would require eight phrases, each of which points to four measures, each of which performs eight events, assuming 4/4 time. This genetic hierarchy reflects the hierarchical nature of music.

Fitness values for the individuals in both populations are derived from the feedback of a human Mentor, who listens to GenJam improvise on a tune and types "g" for good or "b" for bad whenever so moved. Typing a "g" increments the fitness values for the currently playing measure and phrase individuals, while typing a "b" decrements those values. On a typical gig, GenJam uses a dozen or so different soloists, each trained ahead of time on tunes in a particular musical style. When the author becomes bored with a particular soloist, he can simply throw it away and train a new one. This fulfills a fantasy motivated by the author's playing in around 1000 jam sessions over the years, often with soloists he would like to have thrown away…

The human Performer collaborates with GenJam when trading fours in chase choruses, where soloists take turns improvising over four-measure segments of the tune. GenJam listens to the human's last four measures using a pitch-to-MIDI converter, maps what it heard to a phrase chromosome and four measure chromosomes, mutates those chromosomes in the last instant of the human's four, and plays them as its response in the next four. In other words, GenJam uses the evolutionary paradigm for real-time melodic development, evolving what it just heard into what it then plays in response.

2 GENETIC OPERATORS

GenJam follows a fairly standard sequence of initialization, selection, crossover, mutation, and replacement when evolving a soloist. However, the design of these operators is far from standard. For instance, initialization of the measure population uses a fractal generator instead of a uniform generator, which produces initial individuals that statistically resemble individuals from a mature, trained soloist [Biles 98].

Selection and replacement in both populations are accomplished using a tournament selection scheme [Biles 94]. Four individuals are selected at random to form a family. The two fittest family members become parents, and two children are created by performing a single-point crossover at a random bit locus biased toward the center of the chromosome. One of these two children is selected for mutation, and the two children then replace the two original family members that were not selected to be parents. In other words, the better individuals tend to survive and have children that replace the worse individuals. In each generation 50% of each population is thus replaced.

GenJam's mutation operators do considerably more than flip the occasional bit, in an attempt to cope with the fitness bottleneck that characterizes IGAs in general and temporal IGAs in particular [Biles 94]. In fact, most of GenJam's knowledge of melodic development is embedded in its musically meaningful mutations. The measure mutations operate at the event level and include transposition, rotation, sorting, inversion, retrograde, and retrograde inversion. The phrase mutations operate at the measure-pointer level and include reverse, rotation, sequencing, and operators that try to combine highly fit measures or try to insure the genetic diversity of the populations.

GenJam's mutation operators and robust representation scheme also play a major role in its success in trading fours [Biles 98]. When listening to the human performer's last four, the pitch-to-MIDI converter makes many errors, but since the target representation is GenJam's chromosome structure, those errors have no ill effect because GenJam will always play appropriate notes. The mutation operators are important here as well in that they develop the human's four in stimulating ways. There is always a competitive edge to chase choruses, and GenJam is a formidable opponent in its ability to capture and manipulate fours. This makes GenJam gigs great fun to play because the interaction with GenJam stimulates fresh ideas, which increases the spontaneity and energy level of the performance.

3 TOWARD A TENTATIVE TAXONOMY

We now propose a set of dimensions by which to categorize GA-based music and art systems as an initial step toward taxonomy. We will place GenJam on each of these dimensions to illustrate the categories.

One simple dimension is whether the system evolves a piece or an agent. Many evolutionary art systems evolve populations of competing images, but GenJam instead evolves an agent that can create any number of individual pieces. Somewhat related to this dimension is the dichotomy of whether individuals map to structures or operators. GenJam's individuals map to melodic structures, while individuals in a genetic programming system, for example, map to operations.

Two population-related dimensions are the number of populations and the use of the population. Most GA-based systems use a single population, but GenJam uses two interrelated populations. Similarly, most GA-based systems search for a single individual that decodes to a solution, but GenJam needs all the individuals in its populations to represent a soloist. In this way GenJam resembles classifier systems.

Two domain-related dimensions center on temporality and the richness of the search space. Music and video are examples of temporal domains, while still images are non-temporal. The richness of the search space refers to the degree to which acceptable solutions are plentiful in the search space defined by the representations. While solutions are generally sparse in domains for which GAs are an appealing approach, GenJam is probably less sparse than most. We would conjecture that artistic domains in general are more solution-rich than the typical optimization domain, where only one or a handful of optimal solutions exist.

As described above, GenJam's genetic operators, especially its mutations, are more knowledge-based than the operators in traditional GAs. This suggests a dimension of genetic operator intelligence, ranging from traditional and blind to knowledge-based and guided. Another characterization for this dimension might be the purity of application of the evolutionary paradigm. GenJam is quite impure!

Three dimensions address interaction with the system. The first is the degree to which a human is needed to provide fitness. In a classic GA a fitness function can be computed, but in a classic IGA, like GenJam, a human must provide fitness values. A second interaction dimension is whether the interaction occurs in real time. GenJam requires real-time interaction both for training a stand-alone soloist with the mentor, and for trading fours with the performer. The mentor's interaction is in the classic role of providing fitness, but when trading fours, there is no fitness, and the performer instead provides the individuals to be evolved in an admittedly trivial population. Once again, the intelligence of the mutation operators comes to the rescue in a situation where fitness simply cannot be determined. If the human plays well, GenJam will respond well.

The third interaction dimension concerns who the user is. Both GenJam's mentor and the performer are artists concerned with creating a jazz tune. However, the author has used the audience as a collective mentor in what he calls audience-mediated performance [Biles 99]. In such situations each audience member uses a feedback paddle, which is red on one side and green on the other. The author typically plays three to five training tunes where only GenJam improvises. While the audience signals its (dis)pleasure by showing green for good and red for bad, the author collects this feedback and types "g"s and "b"s in proportion to the amount of green or red that is visible. After the training tunes, the author plays tunes using the audience's soloist. The degree to which the audience is in charge of the evolution, then, presents an interesting dimension, particularly in a real-time system.

Related to this is the role of evolution in the piece itself. In a typical GA, evolution is a means to an end, a technique to search for a finished piece that emerges from the evolutionary process. On the other hand, the piece could treat the evolutionary process as an end in itself, using evolution as a development or animation technique. In such systems the evolution *is* the piece. When GenJam uses pre-trained soloists on a gig, it uses the results of evolution. However, in audience-mediated performance situations, the audience's interaction with the author via the feedback paddles provides a nice opportunity for performer and audience to connect. Following the training tunes, many members of the audience will typically hold up their feedback paddles while grinning broadly as the author takes his first solo. Thankfully, the author usually sees more green than red!

We'll end with an aesthetic dimension that is likely to be controversial, specifically the accessibility of the system to its audience. In the computer music arena, many pieces try to challenge an audience artistically by pushing the boundaries of what music is or is not. GenJam, on the other hand, tries to be accessible to an audience by meeting its expectations for straight-up jazz. Instead of challenging an audience artistically, then, GenJam attempts to challenge an audience technically by using a computer to perform convincingly in the mainstream. The best complement GenJam can receive might be, "Nice jazz – amazing you got a machine to play that well." In some ways this is a more difficult goal, because GenJam has to be accessible and accepted on the audience's terms, not just the artist's. The endpoints of this dimension, then, might be totally abstract and inaccessible versus mainstream and accessible.

These dozen or so dimensions are hardly exhaustive and represent only an initial attempt to come up with categories that might be useful in discussing evolutionary music and art systems. Hopefully, this workshop will stimulate interest in and discussion of evolutionary approaches to music and art.

References

John A. Biles (1994). GenJam: A Genetic Algorithm for Generating Jazz Solos. In *Proceedings of the 1994 International Computer Music Conference*, ICMA, San Francisco.

J. A. Biles (1998). Interactive GenJam: Integrating Real-time Performance with a Genetic Algorithm. In *Proceedings of the 1998 International Computer Music Conference*, ICMA, San Francisco.

John A. Biles (1999). Life with GenJam: Interacting with a Musical IGA, In *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo.