

Complexity and Music Generation

Adrian Johnson
The MITRE Corporation
1155 Academy Park Loop
Colorado Springs, CO 80910
1.719.238.2525
Adrian42@gmail.com

ABSTRACT

In this paper, I describe three experiments with complex systems concepts and the emergence of music from noise. The first program uses a chemical reaction simulation to create musical interactions between progressions in several musical keys and cellular automata (CA) within the regions containing the chemical elements. The second program is an applet which generates music within a single key and outputs the state of the CA system as a phase state plot and as audio. The third experiment, related to work by Bilotta [1], modifies the transition rules of a CA with genetic algorithms (GA) to allow more sophisticated structures to arise. Keeping the four classes of CA identified by Wolfram [2] in mind, a person is able to observe several compromises between convergence and divergence.

These methods of harmonization would be useful in creating accompanying audio in a live performance situation as well as stand alone compositions. By using several applets, animations, and other techniques described by Johnson [3] to visualize CA systems and other music generation tools, I endeavor to share my process for post-modern music generation during live video jockey (VJ) performances as well as through my web site, <http://www.rustycycle.com>.

Categories and Subject Descriptors

F.1.1 [Computation by Abstract Devices]: Models of Computation – *cellular automata*.

General Terms

Algorithms, Experimentation, Human Factors, Theory.

Keywords

Procedural Music Generation, Complexity, Cellular Automata, Genetic Algorithms, Evolutionary Computation, Sonification, Orchestration, Dynamic Soundtrack, Harmonization, Synesthesia, Postmodernism.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'06, July 8–12, 2004, Seattle, WA, USA.

Copyright 2006 ACM 1-59593-186-4/06/0007...\$5.00.

1. INTRODUCTION

Because music generation involves both logic and creativity, it is a suitable domain for exploring techniques to create human-competitive solutions. I wanted to examine whether an evolutionary computation (EC) approach could yield music rivaling compositions created by a human musician. Beyond my interest in the connection between EC and human-like machine creativity I was motivated by aspirations to create commercially viable tools and music.

Automatic music generation is a marketable technology. It is an avenue to address the desire for dynamic soundtracks in computer games. These soundtracks need to be generated in such a manner as to reflect the users' changing situations. The music serves as a sonification of the state of the game. Automatic music generation would also provide a way for solo musicians to have interactive accompaniment during live performances. Additionally, music generation tools could be used as educational resources to give budding musicians hints during composition exercises.

A musical work reflects the process with which it was created. Mathematically or algorithmically generated music that retains some aesthetic considerations is often referred to as postmodern music. An aspect of this genre of music is use of a novel approach for composition, perhaps an automated method tuned by hand or constrained in order to meet aesthetic ideals. The novel approach I wanted to explore was automated music generation through use of CA and the notion of cells harmonizing with their neighbors. One can consider a musical progression as a series of state transitions within a cell. One can then consider a collection of progressions (played by real or virtual musicians) as a collection of cells each having state transition rules which are influenced by the state of neighboring cells. In this way the cells can collectively generate harmonization. Local harmonization propagates to global organization. To retain some semblance to twelve-tone music conventions, I placed some constraints on the possible transitions.

An application of machine intelligence that has been popularly studied is chess-playing computers. There are parallels between that subject and music generation because both are a blend between art and mathematics. The computer must improvise counterpoint with a human opponent in chess. In automated music orchestration the machine must parry the human's melodic moves and anticipate future decisions, inferring information from all the pieces on the board – all the notes in the composition.

2. CHEMISTRY OF HARMONY

The experiment I created initially uses a CA system to model a chemical reaction. The interaction of chemicals drives a music

generation and harmonization system. Each quantity of a chemical element in the system represents an amount of a particular key in the arrangement. As the chemicals combine, the products are elements whose corresponding key is a compromise between the keys of the reactants. To represent the “musical elements” I mapped the twelve keys to a twelve tone (tertiary) color wheel. In this way the reaction can be observed more easily; C maj (yellow) and E maj (red) combine to form D maj (orange).

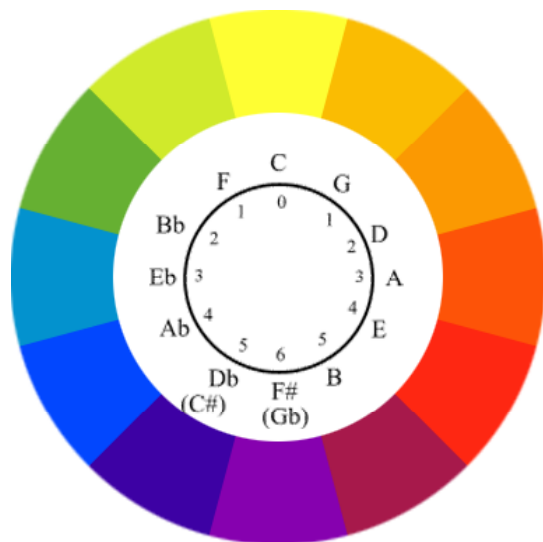


Figure 1. Mapping of key to color.

Reactions are governed by limiting reactants and parameters which specify the ratios of products formed from the reactants consumed. The system uses attractors for each element in order to simulate flow currents in the chemical solution. These attractors can affect each other to create more physically correct and unified behavior, or each attractor can have randomly modulated characteristics to create turbulence in the system. Several thermodynamic aspects of chemical reactions were omitted from the simulation. There are parameters for global growth and decay of the system, as well as growth and decay at the point of reaction. Random quantities of random elements are periodically injected into the system, thereby tuning the amount of potential dissonance.

Addition of the elements and the currents in the fluid stimulate combinations of elements and formations of key gradients. The creation of products between pools of two elements allows a gradient to form. This gradient represents a smoothing in the collection of musical notes within each key, which lessens dissonance and allows the system to sound more cohesive. Each element's cells have associated note values dictated by a chord progression per key. The chord is chosen based on typical progression rules with chord classes (III > VI > II || IV > V || VII > I). The element's chord can remain at the current chord class level or be chosen from the next chord class. Once the chord is determined the cells are mapped to a note in the triad. Leading tone notes can also be added. A cell's note is biased towards being the chord root, but may be the third, fifth, or leading tone based on the values of neighboring notes. This local interaction causes global patterns to emerge. In place of neighbor based selection of

notes, the non-root notes can be randomly placed for a sort of harmonic dithering. In the visualization's note representation mode, notes are colored chromatically; C is yellow, C# is yellow-orange, D is orange, and so on. Figure 2 is an example of dithering over the 50 x 50 cell system, seen in note representation mode.

In creating this music generation scheme I was reminded that the physics didn't have to closely approximate reality, and that a departure from convention may result in interesting behavior. I experimented with a sequence of products and reactants that allowed for a sort of mobius strip – that is, a series of reactions occurred in which products became the new reactants for further reactions, until the initial reactants were regenerated by some reactions. In such a case the chemical reaction aspect of the CA would wrap around on itself, creating some surprising output.

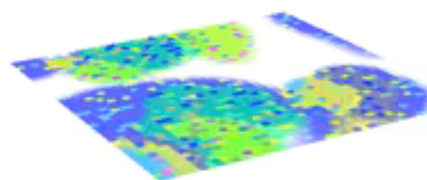


Figure 2. Musical chemicals dispersing in solution.

3. CONVERGENCE AND SYNCHRONY

The second experiment consists of a regular grid of cells, each cell's value corresponding to a chord within a single key. A cell's next value is chosen based on what chord is most prevalent in the four adjacent cells. The cell's state will progress to the chord that follows the most prevalent neighbor state. It takes at least two neighbor cells with the same chord to influence a cell. To allow convergence, there is no case for when a cell has two neighbors with one value and two neighbors with a difference value; whatever neighbor is addressed first in the conditional statement exerts influence over the cell. This causes the cells to sync with the favored chord transition, which then causes more of the system to oscillate together.

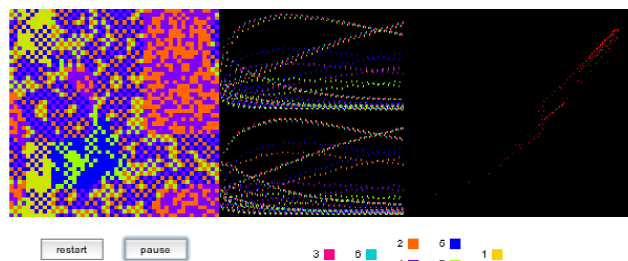


Figure 3. Visualization of emerging harmonic patterns.

In Figure 3, the leftmost area of the applet represents the 50 x 50 cell CA system; the colors are given for the chord each cell contains. The two plots in the center area represent the chords which are most prominent (upper plot), and the number of cells whose chord is the dominant force on another cell (lower plot). The rightmost area represents the phase space. The X-axis

represents the difference between the number of cells containing the most neighbor-dominating chord and the least affective chord, while the Y-axis represents the difference between the least prevalent chord and most prevalent chord. As the cells become synchronous, the clusters of points become tighter in the phase plot. Figure 4 is a sequence of images showing the emergence of greater harmony due to synchrony. I captured the images during corresponding points in different cycles to show the growth of synchronous regions.

Rather than being an entirely deterministic system, this applet's cells are randomly allocated to the II and IV chords when the transition to the appropriate chord class occurs. From there, the progression is deterministic. Rather than a progression pattern such as III > VI > (II || IV) > (V || VII) > I the pattern is III > VI > (II > V) || (IV > VII) > I. This small degree of noise introduced by the probabilistic transition at the *or* allows the systems to have some interesting behaviors along the borders of cell groups that are in sync. The first program has a greater degree of randomness to it, while this system will eventually come to a point where the chords fluctuate very little.

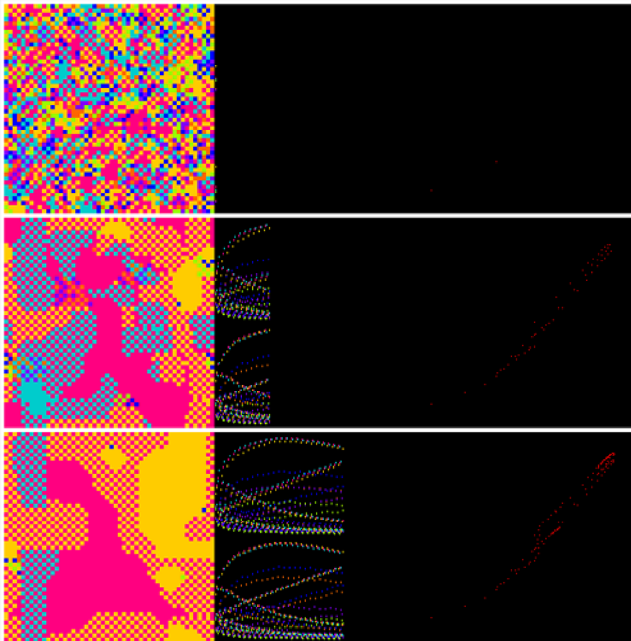


Figure 4. Emerging Synchrony (CA and phase view).

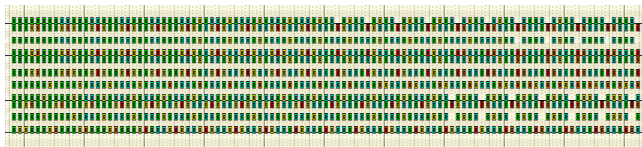


Figure 5. Emerging Synchrony (MIDI velocity view).

To express the automata system in an audible, normalized fashion, I generated MIDI data in which the number of cells at a particular chord state corresponded to the velocity of the appropriate notes. In Figure 5 (using Logic Audio to view the MIDI data) one can see velocity gradually rise from green to red, red being velocity =

127. As more of the cells sync together the notes go from being evenly distributed to having a visible and audible structure.

4. SPAWNING PROGRESSIONS

The third experiment was created to extend the second program in several important ways: to allow harmonization with a live musician, to abstract away from twelve tone music, and to include a genetic algorithm that disrupts convergence. By having clusters of cells whose values are based on user input and whose neighbors will react to resolve dissonance, the local harmonization rules cause a global orchestration to occur. By regarding cell values more abstractly and defining degrees of compatibility between various value-pairs, the cells could represent notes, chords, instruments, or tracks within a mix. In the case of cells representing tracks, the compatibility rating could reflect rhythmic, frequency, instrumentation, or melodic differences (or an aggregate of these attributes). The GA crossover and spawning functions operate by optimizing the CA system for maximum compatibility across all temporal points in a progression while the crossover function serves secondarily to create turbulence and prevent stagnation. In a sense a dynamic equilibrium may be reached, but the magnitude of turbulence allows for interesting variations in the system.

Each cell has a value at each of the eight progression points, representing a transition to make at each of eight bars of music. Again, these values may be assigned to note (micro) or track (macro) structures; the important fact is that there is a degree of compatibility defined for each possible combination of values. The CA system traverses the eight bar progression in unison. At each bar, each cell gains “compatibility points” (CPs) based on the degree of compatibility between the cell and each of its four adjacent neighbors. These CPs are added to a total amount, CP_T . When a cell and a neighbor exceed a CP spawning threshold, CP_S , the two cells breed. The crossover function uses half the alleles from one parent and half from the other using randomly determined allele positions. Mutation occurs at a very low rate, $P_m = 1/10,000$. Two child cells with $CP_T = 0$ replace the parents and the process continues. This mechanism allows for overlapping populations of cells because the GA crossover and spawning occur only when CP_S has been surpassed by a cell and its neighbor.

The CP_T captures the amount of compatibility a cell has over the entire eight bar progression. To know the compatibility at the current point in the progression I consider only the CPs gained in the current bar. This immediate change in CP value, ∂CP , is used as a mask against the cells' musical content. Each cell outputs audio based on the musical value of the cell (whether that be note ID or track ID) multiplied against the gain value $\partial CP / CP_S$. In this way only the cells that are harmonious with their neighbors are audible. This allows the disharmonious cells to lay dormant until the GA causes a beneficial crossover or until a new neighbor value evokes compatibility. Because a cell can gain negative CPs due to extreme dissonance, it may be difficult for a cell to have a chance to breed and alter its alleles. A period of manual tuning was required to determine the amount of CPs warranted by various combinations of cell values.

Figure 6 represents the eight bars of the progression after the initial configuration. Regions of the top row of cells in the center of the CA are assigned progression values by the user. This input is a triad, a typical three-element chord. Remaining cells have

random progressions. In the middle and bottom rows black indicates a value of zero while bright blue indicates the highest values. ∂CP for each bar is shown in the middle row of squares. Bars 4 and 8 in the middle row can be seen to have weighted ∂CP values; since those bars are most important to the musical expression, the ∂CP values are scaled accordingly. The bottom row shows the cumulative CPs.

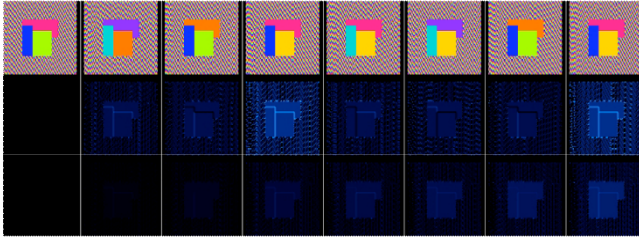


Figure 6. CA bars 1-8 at iteration 0.

In Figure 7 we can observe the 100 x 100 cell system 6400 iterations later. The user input has been distorted and dispersed by the GA crossover. Strips of bright blue indicate regions of greatest spawning activity. The music created at those cells will be most audible.

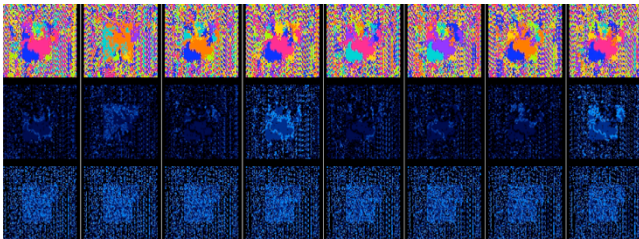


Figure 7. CA bars 1-8 at iteration 6400.

In Figure 8 the CA grid at a single bar is represented by each XY plane, the vertical Z-axis represents the temporal dimension with the current iteration nearest the top. In essence the history of the CA is extruded downwards. The leftmost rendering shows cell music values (color channel) with the gain mask applied (alpha channel). The center rendering is the gain mask, or ∂CP , by itself. The rightmost rendering shows differences in subsequent ∂CP cycles (used to detect oscillation). Viewing the CA with the temporal and masking dimensions became a volume visualization task. Visualization was implemented using the Java OpenGL bindings (JOGL).

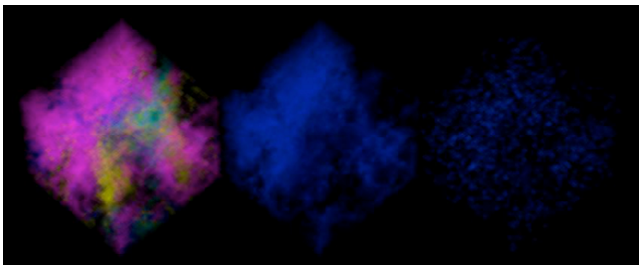


Figure 8. Masked cell values.

5. CONCLUSIONS AND FUTURE WORK

Through a CA for chemical dispersion, a CA with transitions determined solely by neighbors, and a CA altered by GA, I was able to explore several methods of music generation with varying degrees of convergence. In each approach the visual representation of the system allowed for quick debugging and tuning of parameters. I observed having the multi-sensory output – both visualization and sonification – allowed for a better understanding of the systems’ behavior. This work could be extended to sonification of complex systems. Any of these CA-based techniques could also be applied to video mixing. By simultaneously executing video and music mixing one could create artificial synesthesia.

The third approach in which I combined CA and GA allows for the best method for interaction with a live musician. The first two approaches are adaptive but become inflexible, similar to how some neural networks can become too rigid once trained. I would extend this work by having a second chromosome to store progression values. This would allow use of dominant and recessive traits that would make the system more able to adapt to a rapid change in user input. If the live musician changed modalities often within a song, perhaps more than two chromosomes would be appropriate.

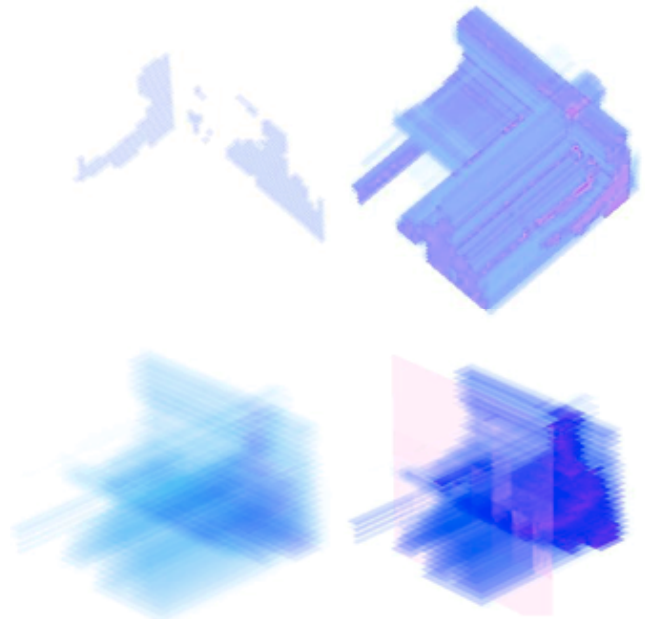


Figure 9. Preliminary CV interaction.

To tune the CA and manipulate cell progressions during live performances I would implement some computer vision (CV) techniques so that I would not be tethered to a computer. By using coarse natural gestures I hope to adjust global parameters such as gain and CP_s . Figure 9 shows preliminary work using two orthogonally mounted web-cams to detect my motion, extrude the segmented regions, weight the voxels, and use the intersected volume as the volume to trigger events (such as reaching through a vertical plane). I also would like to explore using real-time video of the audience to alter behavior of the CA and GA. EC may provide a means to create a CA-based artificial musician to

improvise with me, and also a system with few enough parameters that I can adjust the system's dynamics via video input.

I will explore how notions of multi-dimensional harmonization or compatibility and interaction of CA with GA can be extended to other domains, such as portfolio optimization, computational neuroscience, Department of Defense topics, and network traffic shaping. In many domains the spatial and temporal concerns of music composition will have an analogue. For example, how can we address quality of service (QoS) over network links by implementing differentiated services (DiffServ) approaches that treat router buffers as one-dimensional CAs in which packet priority levels are encoded in cell values? In a given portfolio, what additional investments can be used to orchestrate with current holdings, and how can we predict future market conditions – melodies – in order to invest harmoniously with them?

6. ACKNOWLEDGMENTS

I would like to thank my thesis advisor at the University of Colorado, Dr. Sudhanshu Semwal, for his feedback and enthusiasm during this pre-thesis work.

7. REFERENCES

- [1] Bilotta, E., Pantano, P. and Talarico, V. Synthetic harmonies: an approach to musical semiosis by means of cellular automata. In M.A. Bedau, J.S. McCaskill, N.H. Packard and S. Rasmussen, editors, *Artificial Life VII*, August 2000. Cambridge, Massachusetts. The MIT Press.
- [2] Wolfram, S. Universality and complexity in cellular automata. *Physica D*, 10:1-35, January 1984.
- [3] Johnson, A., Semwal S.K.. Music as an Input Device. IEEEVR2004 proceedings, Beyond Wand and Glove workshop CD-ROM.