

Evolutionary Computation applied to Algorithmic Composition

Artemis Moroni

UNICAMP/NICS

CP 6162

SP, Brazil, 13081/970

Artemis@ia.cti.br

Jônatas Manzoli

UNICAMP/NICS

CP 6166

SP, Brazil, 13081/970

Jonatas@nics.unicamp.br

Fernando Von Zuben

UNICAMP/FEEC

CP 6101

SP, Brazil, 13081/970

Vonzuben@dca.fee.unicamp.br

Ricardo Gudwin

UNICAMP/FEEC

CP 6101

SP, Brazil, 13081/970

Gudwin@dca.fee.unicamp.br

Abstract. This paper presents an end-user interface that allows real time parametric control of sound events. It is an interactive environment in which Evolutionary Computation is applied to Algorithmic Composition. This system uses genetic algorithms to generate and evaluate a sequence of chords played as MIDI data. Melodic, harmonic and voice range fitness are used to control musical features. Based on the ordering of consonance of musical intervals, the notion of approximating a sequence of notes to its harmonically compatible note or tonal center is used. This method employs fuzzy formalism and is posed as an optimization approach based on factors relevant to hearing music.

1. Introduction

Evolutionary Computation has been successfully applied to control music processes. An application of genetic algorithms to generate Jazz solos is described in (Biles, 1994) and this technique has also been studied as a way of controlling Rhythmic structures (Horovitz, 1994).

As an experimental approach, this paper describes a procedure for algorithmic composition based on the controlled production of chord cadences: a population of chords is properly codified according to the MIDI protocol, and is submitted to evolution by the application of genetic algorithms. A fitness criterion is defined to indicate the best chord at each generation, and this chord is selected as the next element in the sequence to be played.

In what follows, a general description of the main components of the computational environment are presented and a musical fitness based on melodic, harmonic and voice range criteria is defined. Finally, the end-user interface is described in detail.

2. Definitions

Some basic concepts are fundamental to the understanding of the main results.

2.1 Population as MIDI data

An auditory event could be characterized by four parameters: *pitch*, *duration*, *loudness* and *timbre*. These four parameters can be considered as attributes of a population of MIDI data:

- pitch* is represented by the MIDI note table;
- duration* depends on the rhythmic genetic cycle defined in 2.2;
- loudness* is related to MIDI velocity table;
- timbre* is given by the General MIDI instrument definition.

Individuals of the population are defined as groups of four voices. These voices are randomly generated on the interval [0...127], which corresponds to MIDI events. In each generation, 30 chords are produced and evaluated. They are internally represented as a chromosome with 28 bits, composed of 4 words with 7 bits (**Fig. 1**).

1011111	1010111	0010111	0100111
---------	---------	---------	---------

Fig. 1 - The structure of a MIDI chromosome.

2.2 Rhythmic Genetic Cycle

The general architecture of the rhythmic genetic cycle is depicted in **Fig. 2**. It is possible to see on the diagram two cooperative processes in the genetic cycle: one producing notes and the other (the interface) consuming notes. Once the initial population of individuals is created, the fitness of each chord can be evaluated. The fitness function is defined as a composition of three terms: the harmonic fitness, the melodic fitness and the voice range fitness. For the evaluation of the harmonic fitness and the melodic fitness, the consonance criterion is considered.

After the fitness evaluation, typical genetic operators of crossover and mutation are applied to the individuals (Michalewicz, 1996). Once the best chord is selected, it is

put available to be played. The interface, which is looking for new notes, send them to the MIDI port.

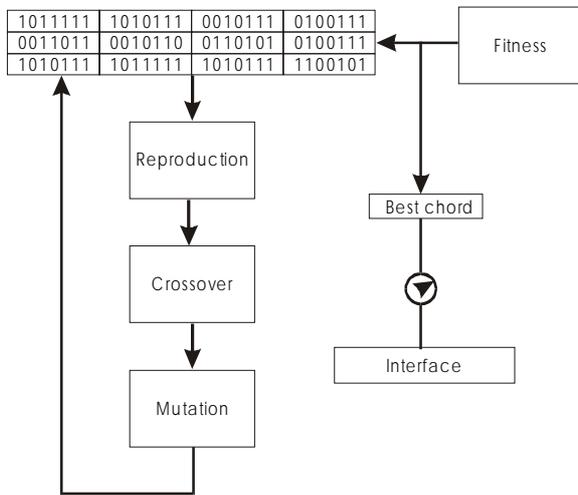


Fig. 2 – The rhythmic genetic cycle

The following steps are realized in the genetic cycle (Pedrycz & Gomide, 1998):

Step 1: Create an initial population randomly;

Step 2: While not stopped (by the user), perform the following:

- Evaluate the musical fitness of each individual in the population;
- Apply the genetic operators to the population of MIDI chromosomes (groups of voices), taken into account based on the musical fitness, to create a new population. That is:
 - **Reproduction:** Copy existing individual strings to a new population;
 - **Crossover:** Create two new chromosomes by crossing over randomly chosen sublists (substrings) from two existing chromosomes;
 - **Mutation:** Create new chromosomes from an existing one by randomly mutating the character in the list;
- **Step 3:** Find the best individual in the new population and play it as an MIDI event. Go to **Step 2**.

The steps above stress the existence of many operations executed in each cycle. The time interval between the selection of the best chords in two successive cycles may be different. On the other hand, the interface is regularly “asking for new notes”. Despite the fact that there is an average time cycle to designate the best individual in each generation, small variations in each time cycle determine the *genetic rhythm*. Different times for the notes being

played are perceived as rhythm of the melody generated by the genetic cycle.

3. Fitness Evaluation

This section presents the criteria used to evaluate the system musical fitness to be attributed to each chord.

3.1 Voices Range Criterion

The voices are associated to the linguistic terms: *bass*, *tenor*, *contralto*, *soprano* and *nh* (no human). The related fuzzy sets are shown in **Fig. 3**. Each voice is assigned to a membership value associated with each linguistic term in the set {NH, B, T, C, S}. For the classification of each voice, the membership function is evaluated for each set and the maximum value is taken. In the case of coincidence, the distance to the center of the fuzzy set is considered. The interval of voices reached by the human voices is assumed to be in the interval $H = [39..84]$, given in MIDI note values.

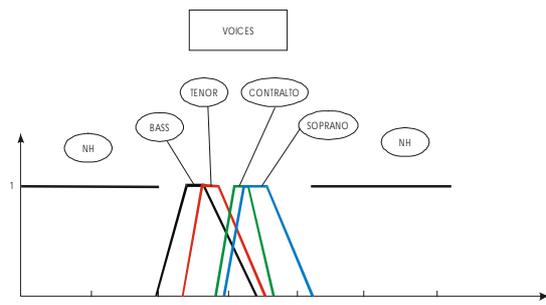


Fig. 3 – The linguistic values associated with the voices

Once the voices of each chord are evaluated according to its distribution in the interval of voices, the voice range criterion returns a value in the set {NH, W, M, G, E}. These linguistic values are associated to the concepts No Human, Weak, Medium, Good and Excellent. The optimal case – Excellent - is considered when the chord contains the voices Bass, Tenor, Contralto and Soprano. In this case, $Nvalues = 4$. The absence of these voices returns NH; the presence of one of them returns W; two returns M; three returns G; with $Nvalues = 0, 1, 2, 3$, respectively. Therefore, the voice range fitness is evaluated as:

$$O = Nvalues/4$$

3.2 The Consonance Criterion

The consonance among the four voices is evaluated as a function of the voice attributes. Using this functional relation, the tonal center cognition can be formulated as an optimization problem. Consonance is a combination of two simultaneous notes judged pleasing to the listener and is defined as a function of the commonality or overlap between the spectral components of notes (Vidyamurthy, 1992). This overlap measurement is then scaled to a value

between 0 and 1, with 1 denoting complete overlap (i. e., the two notes being the same), and 0 denoting no overlap at all. This notion of overlap can be succinctly captured in the fuzzy set formalism further described.

A musical note is a compound tone consisting of its primary tone and upper harmonic partials. Represented graphically as a spectrum, a musical note is a plot of frequency against amplitude. In **Fig. 4** the weighting of the partials of a note versus the relative pitch is represented, and the sum of the weights is normalized. Note that n denotes the n th key on the piano, and that $(n + k)$ denotes the key k semitones above key n . In Fig. 4 and Fig. 5 the upper partials for notes 60 (do) and 64 (mi) were evaluated.

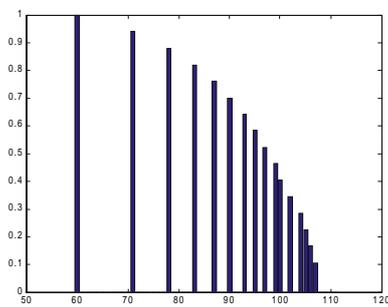


Fig. 4 – The weighting of the partials of note 60 (do)

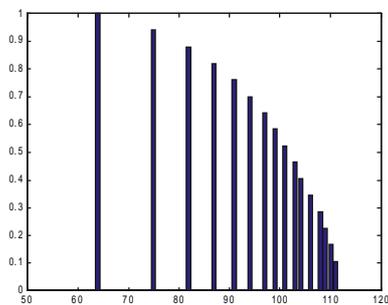


Fig. 5 The weighting of the partials of note 64 (mi)

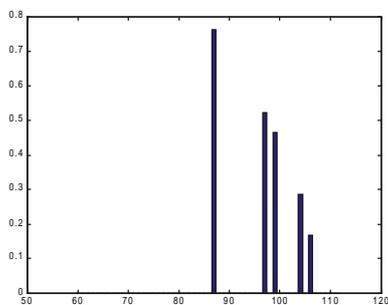


Fig. 6 – The weighting of the *Intersection* between notes 60 and 64

Formally defining, each note is a fuzzy set on a countable universe of discourse. The **consonance** or overlap between the notes is defined as the sum of the intersection of the partial weights (Vidyamurthy & Chakrapani, 1992), and results in a value in $[0..1]$:

$$Co(N_1, N_2) = \sum \text{Intersection}(N_1, N_2)$$

where N_1, N_2 are the sets of the upper partials of the notes. In **Fig. 6** the partials of the *Intersection*(N_1, N_2) is represented. The consonance can be considered the *t-norm* between the sets N_1 and N_2 (the **min** operator is used here) (Pedrycz & Gomide, 1998).

3.2.1 Melodic Fitness

Melody is defined as an ordered sequence of notes with their corresponding start times. In formal terms, a melody can be a string in which each character is an ordered pair (f_N, t) , f_N being the note and t the time duration for playing it. Given an ordered sequence of notes it seems intuitively appealing to call the note that is most consonant with all other notes the coloring or tonal center. Hence the extraction of the tonal center of a sequence of notes would involve finding a single harmonically compatible note, such that the time weighted dissonance between a given note and the other ones in the sequence is minimized. In the Vox Populi system, the consonance is measured according to the value **Id**. This value is obtained from the interface control and can be changed by the user. The melodic fitness is evaluated as:

$$M(N_1, N_2, N_3, N_4) = \text{Max}[C_{Id}(N_j)],$$

$$j = 1, 4;$$

where

$$C_{Id}(N_j) = Co(N_j, Id).$$

3.2.2 Harmonic Fitness

Consonance is a combination of two simultaneous notes judged pleasing to the listener. Harmony is defined here as a function of the overlap between the spectral components of the notes, or the sum of the consonance of the notes present in the chord. Therefore, the harmonic fitness is defined as:

$$H(N_1, N_2, N_3, N_4) =$$

$$\text{Max}(Co(N_1, N_2), Co(N_2, N_3), Co(N_3, N_4), Co(N_4, N_1))$$

3.3 Musical Fitness

The resulting musical fitness is a conjunction of the previous functions and is defined as:

$$F(O, M, H) =$$

$$O(N_1, N_2, N_3, N_4) +$$

$$M(N_1, N_2, N_3, N_4) +$$

$$H(N_1, N_2, N_3, N_4)$$

In the selection process, the chord with highest fitness is selected and played.

4. Vox Populi System

The system was implemented to perform a series of sound experiment and eventually to be used as a tool for Algorithmic Composition.

4.1 Interface and Parameter Control

The user may interfere in the fitness function through five interface controls: 1) *Melodic Control*, 2) *Biological Control*, 3) *Rhythmic Control*, 4) *Octave Control* and 5) *Orchestra Control*; in what follows.

Optionally, a window allows the user to see the voice range, melodic and harmonic fitness evolution, which are shown through bar graphs.

4.2 Real Time User Interaction

The system's real time performance is satisfactory, allowing sound interaction and a possible use of Vox Populi in live music. The program was developed for MS Windows environment and it runs on any IBM Compatible PC with most of the commercial soundboards. Pentium machines will work better starting from a clock of 166 MHz.

The Graphic Interface allows the user to change parameters, interfere in the fitness function and, consequently, on the music evolution.

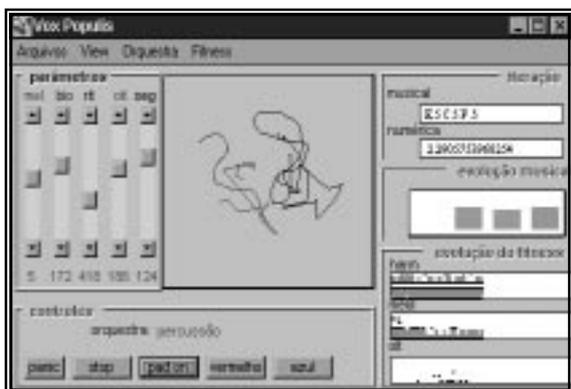


Fig. 6 - Vox Populi graphic interface.

It follows a short description of the controls available to user interaction with Vox Populi.

4.2.1 Melodic Control

The *Mel* scroll allows to modify the value **Id**, which is the tonal center in the evaluation of the melodic fitness.

4.2.2 Biological Control

The *Bio* scroll allows to interfere in the duration of the genetic cycle, modifying the time between genetic iterations. Since the music is being generated in real time, this artifice was necessary to synchronize the different process that are running. This value determines the slice of time necessary to apply the genetic operators, such as crossover and mutation, and can also be interpreted as the reproduction time for each generation.

4.2.3 The Rhythmic Control

The *Rit* scroll changes the time between evaluations of the musical fitness. It determines the “time to produce a new generation” or the slice of time necessary to evaluate the musical fitness of the population. It interferes directly in the music rhythm. Changes on this control make the rhythm faster or slower.

4.2.4 The Octave Control

The *Oct* scroll allows to enlarge or diminish the interval of voices considered in the voices range criterion.

4.2.5 The Orchestra Control

Six MIDI orchestras are used to play the selected chords: 1) *keyboards*; 2) *strings and brasses*; 3) *keyboards, strings and percussion*; 4) *percussion*; 5) *sound effects* and 6) *random orchestra*, that takes any instrument from the General MIDI list. Using the order above, these orchestras are sequentially changed in time segments controlled by the *Seg* scroll.

4.3 Interactive Pad Control

The *Pad On* button enables and disables the pad change on the controls defined before. They can be coupled in two pairs, that can be interpreted as variables of a two-dimensional phase space. This allows a user to draw an oriented curve to determine the music evolution. There are two curves, associated with different colors.

- Red Curve:** describes a phase space of the *melodic* and *octave range* control variables.
- Blue Curve:** describes a phase space of the *biological* and *rhythmic* control variables.

The pad may be musically interpreted as an elementary tool that allows a “master gesture” to conduct the music.

4.4 Fitness Displays

Three other displays allow the user to follow the fitness evolution.

- The first display, on the right top, shows the notes and the fitness of the chord that is being played.
- In the middle, a bar graph shows the four voices (bass, tenor, contralto, soprano) and their value. It is equivalent to the membership function values related to voices' range.
- The last shows the melodic, harmonic and octave fitness plot.

5 Conclusions

Two main approaches were tried for the fitness evaluation and both presented interesting results. The first one, founded in heuristics, provided a less expensive fitness evaluation that allowed to try the system with a larger population, providing more diversification. The fitness of the best chord grew slowly. The later approach, considering the consonance criterion, needs a heavier calculation for the fitness evaluation. In order to assure a good performance of the system the largest population was limited to thirty chords. The resulting composition is less diversified, since the harmonic fitness is more tight, and we can follow the fitness evolving to the unison. This fact highlights the notion that in musical composition not only the consonance is desirable, but also the dissonance. By now, we are trying experiments considering the consonance and the dissonance in the fitness evaluation across the time.

The resultant music moves from very pontilistic sounds to sustained chords. It depends upon the duration of the genetic cycle and the number of individuals on the original population. The octave fitness forces the notes to be in the range **H**, assumed to be the range reached by the human voices and associated with the center region of the notes in the piano. But since that several orchestras of instruments are used, this range is too limited for some of them. The original decision to restrict the generated voices to specific ranges was just to resemble human's voices; nevertheless a user can enlarge this ranges using the Octave Control. The interface was designed to be flexible for the user to modify the music been generated, and can be thought as a prototype environment for algorithmic composition.

Further, we are developing an integration of this system with Gesture Interfaces, as a glove to enhance the man/machine interaction with the purpose of allowing a human gesture be the real time controller. This is a natural extension of the Pad Control. This approach was previously applied in the project ActContAct (Manzolini et al. 1998), where an electronic shoe was used in music generation.

Acknowledgements

We would like to thank our fellow student Leonardo N. S. Pereira for developing the routines to evaluate the consonance criterion. This paper was supported by FAPESP (São Paulo State Research Foundation) and CTI (Technological Center for Informatics), and CNPq (process no. 300910/96-7).

REFERENCES

Biles, J. A. "GenJam: A Genetic Algorithm for Generation Jazz Solos". Proceedings of Computer Music Conference (ICMC '94), pp. 131-137, 1994.

Horovitz, D. "Generating Rhythms with Genetic Algorithms". Proceedings of Computer Music Conference (ICMC '94), pp. 142-143, 1994.

Michalewicz, Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, pp. 283 – 287, 1996.

Manzolini, J., A. Moroni and C. Matallo. "AtoConAto: new media performance for video and interactive tap shoes music", 6th ACM International Multimedia Conference, pp. 3, 1998.

Pedrycz, W. and Gomide, F., "An Introduction to Fuzzy Sets Analysis and Design", The MIT Press, 1998.

Vidyamurthy, G. and Chakrapani, J., "Cognition of Tonal Centers: A Fuzzy Approach", Computer Music Journal, The MIT Press, 16:2, 1992.