Evolutionary Music Composer

Yaser M.A. Khalifa, Hunter Shi, and Gustavo Abreu

Electrical and Computer Engineering Department, State University of New York at New Paltz
New Paltz, 75 South Manheim Boulevard,
New Paltz, NY 12561
(yaserma@engr.newpaltz.edu)

Abstract: In this paper, an autonomous intelligent music composition tool was developed using Genetic Algorithms. The research has been structured into two phases, each of which builds upon the previous one. The first phase of the project was to develop more sophisticated fitness measures for the genetic algorithm, with the goal of applying data compression techniques to identify musically sound patterns in music through music theory principles. In the second phase, methods to use weighted permutations of different fitness functions and generated motifs were explored. These combinations were evaluated and as a result, musically fit patterns were generated. Four musical phrases are generated at the end of each program run, each phrase consists of eight measures, and each measure is one motif of up to eight notes. The generated music piece will be translated through an additional algorithm into Guido Music Notation (GMN) files for further evaluation and alternate representation (midi). The Evolutionary Music Composer (EMC) was able to create interesting pieces of music that were both innovative and musically sound.

1 Introduction

It wasn't until 1026 that Guido d'Arezzo, an Italian monk, proposed to formalize music composition. D'Arezzo resorted to using a number of simple rules that mapped liturgical texts in Gregorian chants, due to the overwhelming number of orders for his composition. Classical composers such as Mozart, Hyden, and Bach used an algorithmic decision process called "Wurfelspiel" to compose minutes and other work. The music was constructed by means of random selection of segments from a table of motifs.

In recent years, and with the development of modern electronic instruments, the computer represents a technological tool to study music in a way that was not possible in the past. Several approaches to composing music with the aid of a computer have been developed [1-5].

2 Genetic Algorithms Implementation

GA is a stochastic combinatorial optimization technique [6]. It is based on the evolutionary improvement in a population using selection and reproduction, based on fitness that is found in nature. The GA operates on a population of chromosomes, where each chromosome consists of a number of genes, and each gene represents one of the parameters to be optimized.

2.1 Chromosome Structure

The composition of music is done in two phases. In phase one motives are generated. A table of 16 best motives is constructed to be used in phase two. These motives will be used both in their current and transposed locations to generate musical phrases in phase two. One gene of the chromosome structure used in phase one is shown in Fig.1. Each chromosome will contain eight genes, allowing a maximum of 8 notes per motif. Each motif is limited to 1 whole note.

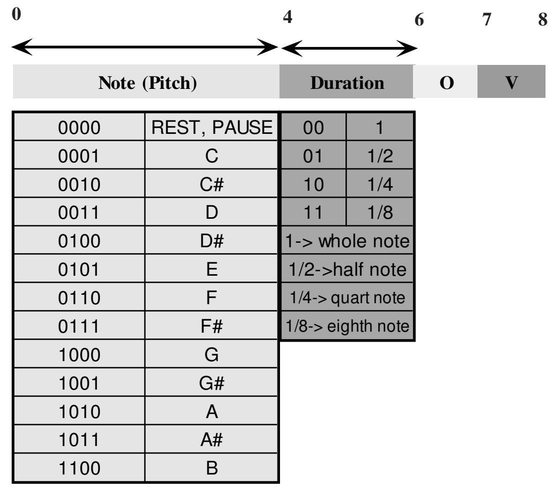

Fig. 1. Gene structure for phase I

Note: Limited to 13 possible outcomes: pause, C,C#,D,D#,E,F,F#,G,G#,A,A#, and B.
Duration: Controls how long a note will be held. Limited to four time measures: 1/8, ¼, ½, and whole notes.
O: Octave, Limited to two octaves: low and high.
V: Velocity piano (soft) and forte (loud).

In phase II, the chromosome consists of two phrases A, and B. Each phrase is eight measures, and each measure is one whole-note motif, see Fig. 2. The motifs are chosen from the motif look-up table constructed in phase I of the program, as in Fig. 3.

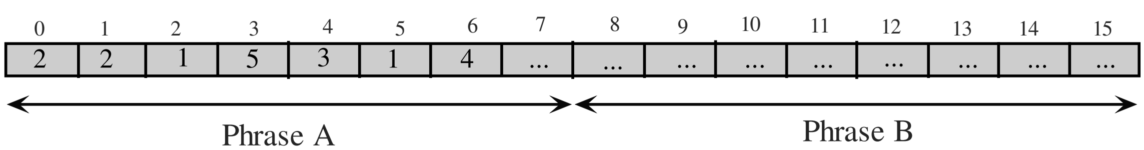

Fig. 2. Chromosome structure for phase II

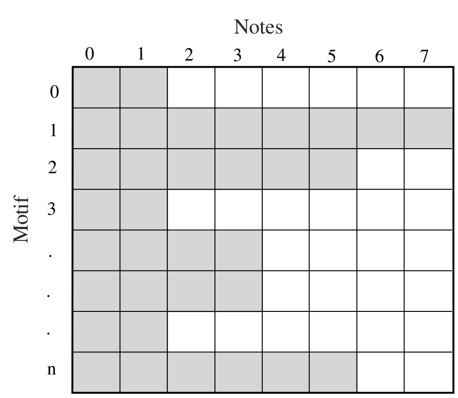

Fig. 3. Motif Look-up table generated in Phase I.

2.2 Evaluation Functions

Intervals Evaluation Function

Within a melody line there are acceptable and unacceptable jumps between notes. Any jump between two successive notes can be measured as a positive or negative slope. Certain slopes are acceptable, others are not. The definitions of the different types of slopes are:

> *Step*: a difference of 1 or 2 half steps. This is an acceptable transition.
> *Skip*: a difference of 3 or 4 half steps. This is an acceptable transition.
> *Acceptable Leap*: a difference of 5, 6, or 7 half steps. This transition must be resolved properly with a third note.
> *Unacceptable Leap*: a difference greater than 7 half steps. This is unacceptable.

As observed from the information above, leaps can be unacceptable in music theory. We model this in GA using penalties within the interval fitness function.

Unacceptable Leap: Penalty = $\sum |1st\_note - 2nd\_note|$ this is calculated over every unacceptable leap.

Acceptable Leap: Penalty = $\sum |(3rd\_note - 2) - acceptable\_error|$

OR

Penalty = $\sum |(3rd\_note + 2) - acceptable\_error|$.

There is also a possibility of bonus within the interval section. Certain resolutions between notes are pleasant to hear, but are not necessary for a good “melody”. These resolutions therefore receive a bonus. Dealing with steps in the chromatic scale, we can define these bonus resolutions as the 12 to 13 and the 6 to 5 resolutions. The 12 to 13 is a much stronger resolution, and therefore receives a larger weight. In total, the bonus will not exceed 10% of the total fitness. Here is how they are calculated:

12 to 13 bonus: $\frac{\#occurances}{15} * .066$

6 to 5 bonus: $\frac{\#occurances * .5}{15} * .034$

The total interval fitness:

$$\text{Interval Fitness} = \frac{1}{total\_error(1 - total\_bonus)}$$

Ratios Evaluation Function

The basic idea for the ratios section of the fitness function is that a good melody contains a specific ideal ratio of notes, and any deviation from that ideal results in a penalty. There are three categories of notes; the tonal centers that make up the chords within a key, the color notes which are the remaining notes within a key, and chromatic notes which are all notes outside a key. Each type of note is given a different weight based on how much a deviation in that portion of the ratio would affect sound quality. The ideal ratios sough were: Tonal Centers make up 60% of the melody; Color Notes make up 35% of the melody; and Color Notes make up 5% of the melody.

Each type of note is given a different weight based on how much a deviation in that portion of the ratio would affect sound quality. Because chromatic notes have such a large weight, there is an alternative weight associated with a cut off point. This is to prevent lost data in the case of large chromatic error. In the interest of clarity and to cleanly display our equations we will have variable declarations. The ideal ratios are taken as follows; Tonal Centers take up 60% of the melody, Color Notes take up 35% of the melody, Color Notes take up 5% of the melody.

We calculate tonal error as follows:

$$\text{Tonal Error} = \frac{|T - IT|}{Z} * a$$

Where a is the tonal center weight, T is the total tonal time, IT is the ideal tonal time and Z is the total time.

And the color error:

$$\text{Color Error} = \frac{|C - IC|}{Z} * b$$

Where b is the color note weight, C is the total color time, IC is the ideal color time.

Calculating chromatic error is a done in a similar fashion.

$$\text{Chromatic Error} = \frac{H - IH}{Z} * e$$

Where e is the chromatic note weight, H is the total chromatic time, IH is the ideal Chromatic time.

Then the total error:

Total Error = Tonal Error + Color Error + Chromatic Error

Finally the ratio fitness:

Ratio Fitness = 1 – Total Error

3 Results

Fig. 4 and 5, are samples of the motives generated in phase I of the program. At the end of phase I, a look-up table of “good” motives is generated. In phase two, a musical sentence composed of four phrases is produced in the form. Only two phrases A and B are genetically generated, then a combination of $ABA^{\#}A$ is produced as the whole piece. Interval and rations evaluation functions were used to connect motives together in the same way notes were evaluated in Phase I. Figure 6 shows one example of a completed piece that is consisted of a combination of $ABA^{\#}A$.

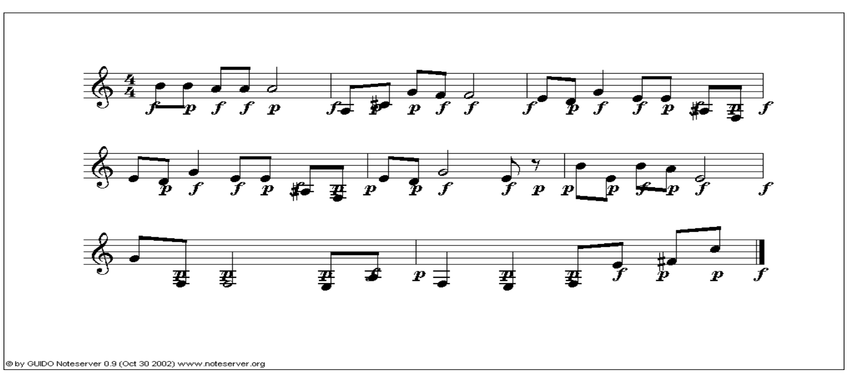

Fig. 4. A motif generated in Phase I

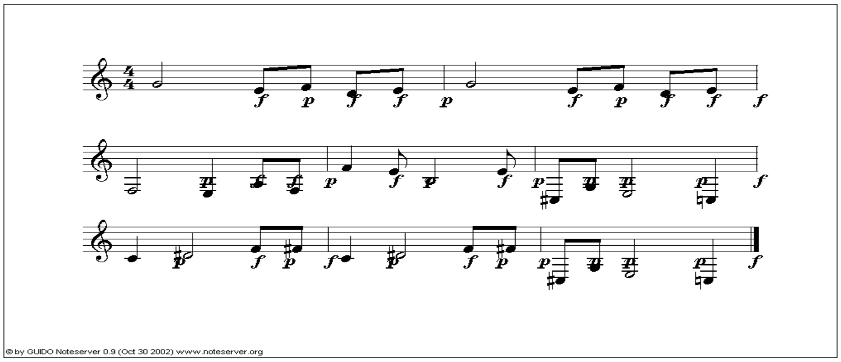

Fig. 5. A motif generated in Phase I

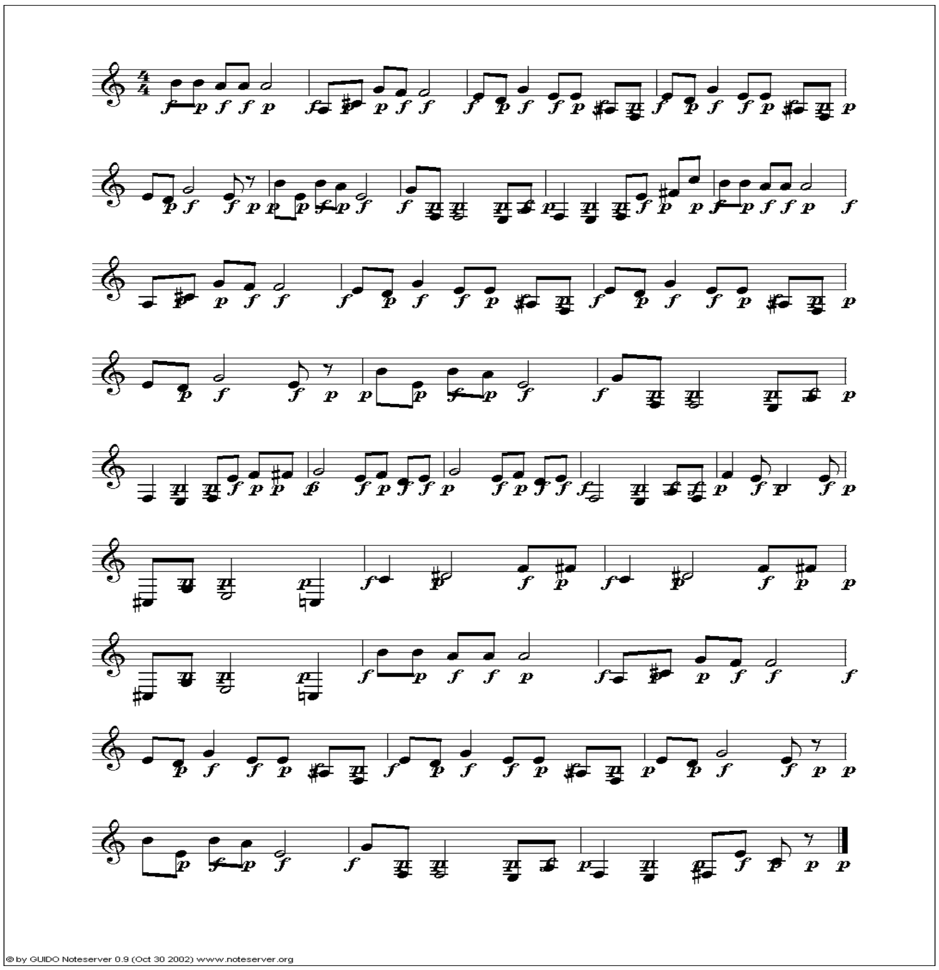

Fig. 6. Complete piece composed of ABA$^{\#}$A phrases.

4 Discussion and Future Work

Although the motives generated by the current program are of reasonably good quality, their combination to form phrases A and B are not as successful. New techniques in evaluating combinations of motives are needed. One approach that will be investigated is the application of formal grammars. A formal grammar is a collection of either or both descriptive or prescriptive rules for analyzing or generating sequences of symbols. In music, these symbols are musical parameters such as notes and their attributes.

In a multi-objective optimization problem such as music composition, different evaluation functions are applied and contribute in the fitness measure of a generated piece. The two main functions that have been designed over the last two years are intervals and ratios. They have been equally considered in evaluating the evolutionary generated music so far. Different weighing methods for various evaluating functions is expected to effect the quality of the resulting music. These could also be effected by type of music sought e.g. classical, Jazz, Blues, etc. In the second phase of the project, methods to use weighted combinations of different fitness functions, or composition rules, will be explored.

References

[1] A. R. Burton, and T. Vladimirova. *Applications of Genetic Techniques to Musical Composition*. Computer Music Journale, vol. 23, 1999.

[2] D. Horowitz. *Genertaing Rhythems with Genetic Algorithms*. Proc. of the 12th National Conference on Artifical Intelligence, AAAI Press, 1994.

[3] M. Marques, V. Oliveira, S. Vieira, and A. C. Rosa. *Music Composition Using Genetic Evolutionary Algorithms*. Proc. of the Congress of Evolutionary Computation, vol. 1, 2000.

[4] Y. M. A. Khalifa, K. Afshari, and B. R. Fogarty. *Genetically Evolvable Music Composition*. Proc. of Northern State University Undergraduate Research Conference, Aberdeen SD, April 20-21, 2001.

[5] A. Pazos, A. Del Riego, et al. *Genetic Music Compositor*. Proc. of the Congress of Evolutionary Computation, vol. 2, 1999.

[6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading MA USA, 1989.