Creating Musical Variations Using Genetic Algorithms

Onkur Sen
12th Grade
onkursen@gmail.com
Oklahoma School of Science and Mathematics
1141 North Lincoln Boulevard
Oklahoma City, OK 73104

Abstract

The space consisting of musical compositions is broad and expansive. Individuals have particular preferences to certain types of compositions; thus, generating creative variations can produce new, interesting music. Although human-based approaches are prevalent, they are biased by the designer's preferences and do not necessarily conform to the desired standards of any particular individual. However, computational methods of generating and evaluating music, especially genetic algorithms (GAs), have been shown to provide a significant improvement in the innovation and variation of the generated pieces. We propose a genetic algorithm that utilizes seeded stochastic search, i.e., one or more sample pieces of music are used as starting points for probabilistic exploration of the space of musical pieces. We then evaluate the generation of novel variations of selected samples with different genetic representation of scores, fitness metrics, niching methods, and recombination operators.

1 Introduction

Music has been an integral part of humanity from the beginning of civilization. It originated in forms such as folk tunes and religious hymns and developed differently in various regions, e.g., European as opposed to Eastern music. However, it was not until the 15^{th} and 16^{th} centuries with the rise of Western classical music that it was necessary to develop a representation that was consistent among composers across Europe such that music could be distributed in a widespread fashion. These efforts in standardization focused on developing a single form of notation for transcribing music that took numerous parameters of both the notes (e.g., pitch, rhythm, and ornamentation) and the entire musical work or its sections into account (e.g., tempo, key signature, time signature, and dynamics).

In addition, the development of musical variations based on a theme has been popular since the Baroque musical period. Each variation changes the theme in some consistent manner which sets the variation apart from the theme as well as other variations. Examples of famous variations in Western classical music include Johann Sebastian Bach's *Goldberg Variations*, Ludwig van Beethoven's *Diabelli Variations*, and Sergei Rachmaninoff's *Rhapsody on a Theme by Paganini*. Variations are not limited to classical music, however; they are prevalent in contemporary music as well in the form of remixes and edits.

The advent of computer-generated music takes advantage of this standardization by encoding the music into some sort of representation scheme, which allows the music to be manipulated. There have been numerous methods of generating computer music, but one of the most successful methods involves using genetic algorithms, which have proven to be successful in solving complex problems in other areas of computer science. This special type of evolutionary algorithms functions by taking a target individual and generating a population of random individuals and breeding them over many generations to create individuals which are similar to the original. We are particularly interested in generating musical variations by providing one or multiple samples and using genetic algorithms to create an innovative musical selection which is collectively similar to the original samples.

The rest of the paper is organized as follows: in Section 2 we briefly overview relevant literature, in Section 3 we present our problem formulation and outline our experimental framework and procedures; in Section 4 we analyze our experimental results and their implications; in Section 5, we form a conclusion about our results and propose future work.

2 Related Work

Algorithmic music composition has been popular since the 1990s, and genetic algorithms have been shown to be particularly useful for providing innovative solutions for generated musical selections. Burton and Vladimirova provide a comprehensive overview on genetic techniques, including genetic algorithms and genetic programming (1999). One criticism of genetic algorithms addressed is that the search space for the algorithm is vast. Thus, constraints, such as limiting the range of notes to choose from, provide a limitation on the search space and allow the algorithm to function more efficiently. In addition, Mahfoud provides a detailed study on niching, which provides more innovative solutions by forcing a certain amount of variation (1995). Johanson and Poli provide a method of generating music by means of genetic programming which is then rated by humans (1998). Our contribution is to combine genetic algorithms with fitness sharing, a type of niching, as well as various genetic operators in order to create samples which are unique and innovative variations of the original pieces given; in particular, we would like to examine the effect of providing multiple samples on the quality of the resultant piece from the algorithm.

3 Methods/Experimental Procedures

3.1 Representation of Musical Scores

In order to apply the genetic algorithm on a set of original samples, a consistent method was needed to represent musical selections in a form that can easily be manipulated. We achieved this by breaking each piece down into many **time-slices**, each representing a note of the same duration. The time-slice for any given passage was taken to be the note of shortest duration in the passage, mostly 16^{th} notes. In addition, the pitches of the notes were represented with the values used in MIDI (Musical Instrument Digital Interface), a popular method of representing music for notation. These values of pitch ranged from 0 (a very low C) to 127 (a G that is 10 octaves higher). In addition, we added two special values: 128 for the **hold operator** as well as 129 for a rest. The hold operator serves to indicate that a note is played for a longer duration than the fundamental time-slice will allow. For instance, if one wished to represent an 8^{th} note in a passage where the time-slice was a 16^{th} note, it would be represented by the pitch of the note followed by a hold operator. Thus, if the 8^{th} note was a middle C, the representation would be "48 128".

3.2 Genetic Algorithm

A **genetic algorithm** is a type of evolutionary algorithm that models the Darwinian notion of natural selection as a population evolves by crossing over the fittest individuals and creating offspring which are then used in the next generation of individuals. This causes weaker individuals to perish while leaving the stronger and more optimal solutions in future generations.

3.2.1 Initialization of Population

Given a set S of n original works, we first **equalize** all members of S, i.e., standardize the time-slice among them. This is done by inserting holds after every note in the piece with the longest time-slice. This allows for easy manipulation of the pieces as each location can be directly compared among pieces. In addition, we mutate each member of S with a small probability $\epsilon \approx 0.05$ so as to maintain a certain distance from the original pieces (mutation is discussed in detail in Section 3.2.4).

We subsequently initialize a population of $P = 10n$ individuals. A portion αP ($\alpha < 1$)

individuals are simply pieces with completely randomly generated valid values (i.e., 0-129). The other $(1-\alpha)P$ individuals are variants of the original pieces which are mutated with an average probability.

3.2.2 Fitness and Niching

The **fitness function** f is the method of determining how "good" a test piece is with respect to the original pieces. For comparing a piece to an original piece, the test piece was first transposed to the key of the original solution by finding the values of first notes in the test and original pieces and shifting all the notes in the test piece by the difference in those values (holds and rests are unaffected by the transposition). The fitness then becomes a cosine similarity between the test and original pieces with a slight modification: if the location in question for either piece is a hold, the last note or rest value is substituted for the hold. This prevents the fitness function from becoming representation-dependent and gives a more accurate depiction of how similar the two pieces. This is important because if this construct were not in place, then a held note compared to a note repeated many times would result in a poor value of fitness. Furthermore, it is important to note that the two pieces may be of different length; if this is the case, then the fitness is computed for the length of the shorter piece, and the portion of the longer piece which does not have corresponding locations in the shorter piece is left unchecked.

In addition, we also introduced the concept of **fitness sharing**, which is a type of niching. **Niching** is a technique used as a supplement to genetic algorithms in order to promote variation. This is achieved by creating a new fitness function f' to not only promote similarity to the original works, but to also emphasize dissimilarity among individuals in the population. Thus, the previously obtained fitness is then divided by the summation of a simple cosine similarity c between one individual and each other individual in that generation. Thus, f' of a piece a with respect to an original piece o can be represented as

(Mahfoud 1994):

$$f'(a,o) = \frac{f(a,o)}{\sum_{s \in S} c(a,s)}.$$

Finally, the total fitness of an individual is the average of the modified fitnesses among all original pieces, i.e.

$$f'(a) = \frac{1}{n} \sum_{o \in S} f'(a,o).$$

3.2.3 The Next Generation: Elitism and Crossover

In order to produce the next generation of individuals, we first include the two fittest individuals in the population directly in the next generation. This is a concept known as **elitism**, which keeps some fit individuals between generations. To fill the rest of the population, for each "offspring" individual, two parent individuals are chosen via **tournament selection**. This occurs by twice randomly choosing two individuals from the population and selecting the individuals with higher fitnesses as parents. Thus, although the initial selection of parent candidates is random, fitter individuals are more likely to be chosen as parents. Subsequently, the parents are probabilistically subject to p**-point crossover** which is facilitated by a bitmask which directs the values of the parents. The two offspring resulting from each crossover operation are then added to the population. This is repeated until the number of offspring (including the individuals carried over due to elitism) equals the population size. After a new generation is created, the old one is discarded.

3.2.4 Mutation

Now that a new generation has been created, the individuals each undergo **mutation**, which simply adjusts each note in a piece up or down by a certain distance with a certain probability. This is another way of encouraging innovation although it is not as directed as niching. Rather, the random changing of notes in a piece may lead to the development of interesting harmonies and melodies.

3.3 Experimental Framework

We developed the experimental framework for this project in *Java* utilizing the *Eclipse* Integrated Development Environment (IDE). We gave the genetic algorithm, which ran for 100 generations a sample set consisting of both single and multiple sources. For the single source set, we gave the algorithm three out of J.S. Bach's Two-Part Inventions: Nos. 1, 8, and 13. For the multiple source set, we included a Bach invention as well as a "different" kind of piece, such as "The Swan" by Camille Saint-Saens and "Minute Waltz" by Frederic Chopin. Furthermore, we wished to see the effect of niching on the variation (as well as quality) of the music produced.

4 Results and Discussion

4.1 Single Source

For our experiments with a single source, we provided the genetic algorithm with three of Bach's Inventions: 1, 8, and 13. Figure 1 shows the first two measures of the original Bach Inventions 1, 8, and 13. Figure 2 show the variations generated from Inventions 1 and 8. We notice that there is a slight difference between the variation and the original, and although the tonal quality of the variation may not be as pleasing to the ear as the original, the variation is quite interesting and noticeable. This shows the limitations of our current algorithm, which models music somewhat primitively and must be refined before achieving more acceptable variations. For instance, we see that in the variation for Invention No. 1, a G eighth note in measure 1 is replaced by a G#-G sixteenth note sequence. This is an interesting change in both rhythm and pitch (including the use of occasional accidentals) which would be common in variations composed by a human. However, the eighth note sequence following it (D, A#, C#) is slightly unorthodox and would not be found in a normal variation. In addition, the variation on Invention No. 8 results in a somewhat chromatic, yet repeating descending cadence in the opening measures, which is also an interesting twist.

In addition, we examined the effect of niching on the quality of the piece. Figure 3 shows the variation for Invention No. 1 which includes niching. Although the variation is much more prominent, the piece as a whole is not very cohesive and does not sound as good as that without niching. However, it does have some interesting parts. For instance, the last part of the excerpt in the figure is a descending 16th note cadence which sounds quite eerie. Thus, although niching may produce interesting results, it is not particularly suited for this type of problem which also involves subjective analysis.

4.2 Multiple Sources

The results from giving the genetic algorithm multiple sources were quite interesting and unique (niching was not used). We attempted to see the effect of giving the algorithm multiple like and unlike pieces. Figure 4 shows the variation generated by giving the algorithm all 3 inventions. The result seems to resemble Invention No. 13, but the tone is completely different (the first 3 notes form a major triad), giving a completely innovative solution that sounds quite like an invention. Although the tonal quality is not exceptional, the fact that the variation maintained the style of the piece can be rendered as a success. Figure 5 shows the variation generated by giving the algorithm Invention No. 1 and Chopin's "Minute Waltz," which are two unlike pieces. We see that although the form of the piece is much like an invention, the key of the piece cannot be properly distinguished, and the tonal quality of the piece is not that high. However, it does contain technically interesting sections, such as the alternation between the G# and the F# in the piece, which is quite common in music.

5 Conclusions and Future Work

Our research goal was to determine the effectiveness of genetic algorithms and other genetic operators in creating euphonious musical variations given a sample set. Although we have generated many interesting variations, one thing that is noteworthy in our work is the boundary of our algorithm. As this is our first attempt to systematically represent music, the algorithm is naturally slightly crude. For example, it does not take into account key signature, tempo, ornamentation, and many other intricacies of music. Thus, the models we have generated are but a rough approximation of a much deeper and euphonious possible variations from these samples. However, we have created a way to consistently represent music and generate multitudes of variation (some interesting and euphonious, others not) which serve as interesting innovations.

References

[1] Burton, Anthony R. and Vladimirova, Tanya. 1999. Generation of Musical Sequences with Genetic Techniques. *Computer Music Journal*, p. 59-73.

[2] Mahfoud, Samir W. 1995. Niching Methods for Genetic Algorithms [Ph.D. Thesis]. Urbana (IL): University of Illinois at Urbana Champaign.

[3] Johanson, Brad and Poli, Riccardo. 1998. GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters. Proceedings of the Third International Conference on Genetic Programming, p. 181-186.

[4] Hancock, Peter J.B. 1994. An empirical comparison of selection methods in evolutionary algorithms. Lecture Notes In Computer Science; Vol. 865 (Selected Papers from AISB Workshop on Evolutionary Computing), p. 80-94.

[5] Collins, Tom et al. 2010. Using Discovered, Polyphonic Patterns to Filter Computer-generated Music. *Proceedings of the International Conference on Computational Creativity*, p. 1-10.

6 Figures

Figure 1: Excerpts from Bach's Inventions No. 1, 8, and 13 (top to bottom).

Figure 2: Excerpts from the variation generated from Inventions No. 1 (top) and 8 (bottom) without niching.

Figure 3: An excerpt from the variation generated from Invention No. 1 (with niching).

Figure 4: An excerpt from the variation generated from Inventions No. 1, 8, and 13.

Figure 5: An excerpt from the variation generated from Invention No. 1 and the Minute Waltz.