Creating Classical South-Asian Music through Evolutionary Algorithms

Muhammad Ali Siddiqui – ms06875
Muhammad Azeem Haider – mh06858
Mustafa Sohail - ms06860

March 26, 2024

Instructor: Dr. Saleha Raza

Introduction

Creativity has always had a complicated relation with modern computing. While the modern computing systems have grown to be scarily accurate in prediction, regression, and classification tasks. The question regarding the creativity of a computing system is still unanswered. Would a computing system ever create art as profound as Vincent Van Gogh's *Starry Night*? Given all the resources in the world, can a system come up with a composition as earnest as Beethoven's *fifth symphony*? The best musicians in the world play their instrument with a passion that is unlike anything else, and would a modern computing system be able to recreate the passion it takes in creating a composition that forces the listeners to focus on the music. The following course project aims to explore the problem of implementing an evolutionary process capable of creating classical South-Asian music composed on a **Tabla** based on music theory to ensure melodic results.

Literature Review

The musical score known as the Illiac Suite is generally agreed to have been the first score generated on an electronic computer [**2**]. In the past evolutionary computation systems have been successful in creating music based on music theory rules or music similar to an individuals preference [**1**]. Researchers trying to produce music through evolutionary processes or artificial intelligence soon realized that creating soothing music was a subjective task that required human intervention.

GenJam was one of the worlds first system that produced music through Genetic Algorithms. The user graded generated Jazz phrases as either good or bad. The feedback was then used to build a Jazz solo. The future versions of GenJam used an Artificial Neural Network to act as the fitness function rather than using human as a fitness function. However, the researchers found that the result was somewhat lacking the melodic sound. The conclusion from this was that experience music was complex and subtle which cannot be captured perfectly by statistical models [**1**] [**3**] [**4**].

Previous research in algorithmic composition has explored a variety of methods, often employing fitness functions grounded in melodic and musical theory concepts. One research provides a comprehensive overview, particularly focusing on genetic algorithms (GAs) and their challenges [**9**]. Some implementations involve evolving musical operations from initial sequences, yielding compositions responsive to musical phrases or connecting initial and final patterns [**5**]. Granular synthesis techniques have also been utilized, evolving sequences of sound wave parameters to generate musical compositions, while one research employs genetic programming to construct structured sequences featuring repetitions, transpositions, and other human-like elements commonly found in musical compositions [**6**] [**8**]. To demonstrate the efficacy of evolutionary algorithms in generating diverse and structured musical compositions, a suitable formulation of fitness functions is required.

Determining the fitness function is one of the difficult tasks in genetic algorithms for music compositions. Different researchers have employed different techniques to tackle the problem. The translation of one melody to another can use some form of distance to the melody that is targeted to be created [**1**].

While fitness measures based on the musical collection have been explored extensively. This method also suffers from bias, since the resultant music is limited to creating music similar to the musical corpus.

Techniques from Computational Intelligence

The following project explores the techniques from genetic programming to create our melody. In particular, the project uses the branch of evolutionary algorithm to generate the sound of classical music. Evolutionary algorithms are based on the phenomenon of natural evolution where the main functions are crossover, mutation, fitness function as well as the chromosome representation.

The chromosome is fundamentally a sound note made by combining multiple different tabla notes. Multiple tabla notes can be played at any instance together as well to create more realistic melodies, we will refer to that as notes stacking. The chromosome will store the information of ordering of these notes so they can be recreated to evaluate fitness. The crossover and mutation operators will modify two things; the ordering of these notes and stacking of these nodes at different instance.

It is essential that we design a robust fitness function since our goal is to eliminate human interaction and have the algorithm evaluate the music's quality itself. The project's aim is to draw inspiration from musical theory to capture the essence of the music. Some of the key components that can be included to give a fitness value to each chromosome are:

- Rhythmic Patterns: Making sure the generated beat follows the patterns of South Asian Music.
- Melodic Cohesion: Making sure the music flows well together.
- Dynamic Variation: The intensity of the music i.e the loudness and speed should vary.

These are just abstract concepts for now that will need to be modelled to give a fitness value to the chromosome.

Dataset

The instrument, tabla, is fundamentally composed of different taal sounds. For our project where we are creating music from scratch, we needed these taal sounds to construct together a melody. We will be using the sound files available in the following repository. The repository has more than 200 sound files in **.wav** format which can be used to compose the sound for the project.

Additionally, the project could also explore tabla-js. This is a JavaScript library which is supplied with the **taal notations** as human language input and it generates the sound from those notations. The demo can be seen here for further clarity. There is an option to integrate this library in our code to generate the melody.

Final Outcome

The final product will be presented through an application that will allow the user to interact with our program. Other than the length of the musical composition, there will be a few parameters which will allow the user to control the type of initial generated dataset. We have decided on some tentative parameters which are:

- Tempo: user can adjust the speed/tempo of the music, this will control the table strokes played in unit time.
- Loudness: to control the intensity of the generated sound.
- Complexity: this will allow the user to control the stacking of the table notes and the density of this stacking.

References

[1] Loughran, R., & O'Neill, M. (2020). Evolutionary music: applying evolutionary computation to the art of creating music. Genetic Programming and Evolvable Machines, 21, 55-85.

[2] L.A. Hiller Jr, L.M. Isaacson, Musical composition with a high speed digital computer, in Audio Engineering Society Convention, vol 9 (Audio Engineering Society, 1957)

[3] Biles, J. (1994, September). GenJam: A genetic algorithm for generating jazz solos. In ICMC (Vol. 94, pp. 131-137).

[4] Biles, J., Anderson, P., & Loggi, L. (1996). Neural network fitness functions for a musical IGA.

[5] Alpern, A. (1995). Techniques for Algorithmic Composition of Music.

[6] Fujinaga, I. and Vantomme, J. (1994) ”Genetic algorithms as a method for granular synthesis regulation”, Proceedings of the International Computer Music Conference. pp 138-141

[7] Johnson, C. G. (2012, April). Fitness in evolutionary art and music: what has been used and what could be used?. In International Conference on Evolutionary and Biologically Inspired Music and Art (pp. 129-140). Berlin, Heidelberg: Springer Berlin Heidelberg.

[8] Johanson, B. E. (1997) ”The GP-Music System: Interactive Genetic Programming for Music Composition”, University of Birmingham, Second-Semester Mini-Project Report.

[9] Beckert, D. (1997) ”Algorithmic composition”, BSc thesis, University of Reykjavik