Algorithmic Arrangement for Close Jazz Harmonies

Overview

A musical arrangement usually starts with an existing melody, song, or composition, and produces a new composition which changes certain elements of the original, such as the instrumentation, key, and even harmonies and sometimes the very melody itself. A standard form is the Theme and Variations, in which the composer starts with a tune, the theme, and then generates variations that might vary in key, tempo, time signature, harmony, or other features.

Arrangements often change the instrumentation of a piece. For example, Liszt created virtuoso piano version of the nine Beethoven symphonies. He wanted to be able to perform the symphonies on his own. It is common to create an arrangement of an existing piece to suit the structure of a specific musical ensemble. The marching band literature is full of arrangements of music that was seldom intended originally to be performed by marching bands.

Similarly, the repertoire of a cappella singing is largely derived or arranged from music that was nor originally meant for a cappella groups. Creating arrangements is thus the primary way of expanding the a cappella repertoire.

One style of a cappella arrangements is close jazz harmony. In this project, I propose to automate that process, creating a computer program. The input to the program would be a melody and associated chords, and the output would be a four-part vocal arrangement.

The normal process of arranging and harmonizing a song is laborious and time consuming. However, the process should be amenable to automation based on common jazz harmony, theory, and style conventions. I will create an algorithm for arranging close jazz harmonies and implement it using Euterpea.

Details

Musical arrangements are manually time consuming. Even if you know exactly how you want the music to sound, the act of entering it into musical software, note by note, is sometimes arduous. Automating the process of harmonizing a melody based on established jazz theory conventions would allow music arrangers to find possibilities they may not have thought of and could allow more for experimentation given that producing different types of harmonies would take less time. The output of such a tool could serve as the initial draft for a musical arranger to work off of.
There are several aspects to the harmonic choices. A significant choice is the style of harmonization being used, such as Drop 2 harmony, Andrews Sisters style, or even simple unison. Further, there are harmonic substitution rules that allow for substitution of one chord for another, such as tritone substitutions. In addition, arranging requires various voice leading considerations, since often you do not want voices to cross or to give all the parts large leaps, instead opting for motion by half-or whole-step.

For this project, I would develop a function that would take in a melody to a song, alongside the key of the song, and the changes to the song. It would also allow the user to specify among different types of harmonization styles and techniques. It would then output an arrangement in the form of a midi that could be then listened to and opened in music notation software for modification.

**Timeline of Deliverables**

**Week 1**

Research types of jazz harmonies. Read Mark Levine book on Drop 2 harmonies. Research other styles of harmonization.

**Week 2**

Research jazz voice leading styles and harmonic substitution.

**Week 3**

Develop and adapt data structures for harmonies. Develop and adapt data structures for melodies if something other than Music Pitch needs to be used.

**Week 4**

Identify an initial example song, represent it using the data structures, and manually arrange the song, noting the decisions and choices made along the way. The system should also generate midi output of the music.

**Week 5**

Review the initial process of manually arranging the song, and develop rules that implement the choices that were used.

**Week 6**

Develop data structures and algorithms for implementing the arranging rules such as use of basic voice leading.
Week 7

Test the rules data structures and algorithms on the initial song. Evaluate the arrangement and identify changes that should be made to the system.

Week 8

Identify a second example song, represent it using the data structures, and let the system arrange it. Evaluate the arrangement and identify any further changes that should be made to the system.

Week 9

Vary the chords of the initial songs to explore alternative arrangements. Develop rules that modify chords using jazz harmonic substitution, and harmonic preference rules.

Week 10

Test the system with additional melodies and chords. Refine and expand the algorithms as needed.

Week 11

Miscellaneous final coding and final write up.

Week 12

Develop website pages for cleaner presentation of project.