Chapter 8

Practical Exercise

Algorithmic Composition of Contrapuntal Music

This is an interdisciplinary project that synthesizes concepts of music, physical science and computer science through algorithmic music composition. Thus, the assignment is suitable for an individual who has a background in music theory, digital audio and MIDI, and computer programming. The project could also be undertaken by an interdisciplinary team who combine their various backgrounds.

The assignment is to write and implement an algorithm that creates two-voice contrapuntal music. You can use whatever programming language you choose.

To get you started, we offer a definition of contrapuntal structure.

Contrapuntal Structure

For this project, your goal is to implement a program that writes music in contrapuntal style, since this form is so fundamentally algorithmic. Contrapuntal music, also called counterpoint, is a structure of musical composition that originated in the Renaissance and was developed extensively in Baroque and Classical music, particularly in canons and fugues written by Bach and Mozart. Many species of counterpoint exist, but we will focus here on the fundamental rules common to all of them [1] [2].

Counterpoint consists of two lines of musical notes, also called voices. Their rhythmic, harmonic, and melodic relationships are what give interest, complexity, and beauty to the piece. Each line provides a melody, but the two melodies line up vertically in intervals that follow a prescribed pattern. That is, the notes that line up vertically are chosen from a particular chord such that the chords follow the diagram shown below. (We assume here that you know some basic music theory. For an introduction to the fundamentals, see Chapter 3 of Digital Sound and Music (Burg, Romney, and Schwartz 2016).)

![Figure 1: Rules for chord progressions in contrapuntal music](image.png)
Each Roman numeral in the diagram represents a chord. The arrows show the permitted chord sequences. The piece begins in chord I, the root of the chosen key. The up arrow after chord I indicates that any chord, not including itself, may be used after I. This implies a choice, which can be made randomly in a computer program. Let’s say that chord iii is selected next. The fact that there is an arrow between iii and vi as well as an arrow between iii and the box containing IV and ii indicates that either choice could be made. Let’s say that the next step is randomly determined to be the box with IV and ii in it. Because both chords are in the box, again the system must make a random choice between the two. If IV is selected, it can go to ii, I, or the box with V and vii°. If ii is selected, it can go to I, or the box with V and vii°.

As counterpoint composition progresses, the notes for the two voices are selected from the notes in the chosen chord. For example, in the key of C major, if the next chord is I, the notes C and E, E and C, C and G, G and C, E and G, or G and E could be chosen (order relating to which line gets which note.) Thus, while there are rules, there is also a great deal of choice within these rules.

Additional constraints in contrapuntal composition emerge from the concepts of consonance, dissonance, and motion. In Western music, the notion of consonance is loosely based on the human perception of pairs of notes that are harmonic and "sound good" together. This sensory phenomenon is related to the intervals between the notes and the way in which the cycles of their respective sound frequencies "match up well" over time. Intervals are classified below.

- **perfect consonances**: unison, octave, fourth, and fifth (A fourth is considered dissonant in some contexts.)
- **imperfect consonances**: major or minor third, major or minor sixth
- **dissonances**: major or minor second, augmented fourth, diminished fifth, and major or minor seventh.

Motion relates to the composition as it proceeds in the horizontal direction. That is, motion results from changes in pitch of the two voices. Each step of the two voices from one note to the next can be identified as one of these types of motion:

- **direct motion**: The voices move in the same direction in pitch (up or down). *Parallel motion* is a type of direct motion in which the voices move in the same direction by the same amount.
- **contrary motion**: One voice moves up while the other moves down.
- **oblique motion**: One voice moves up or down while the other does not move.

In contrapuntal composition, the type of motion that is permitted from one note to the next is captured in the following rule: If the chord progression goes from either perfect or imperfect into perfect consonance, direct motion is not allowed. For better musicality, contrary motion is preferred over direct or oblique.
Algorithmic Approach

A number of approaches to algorithmic composition of contrapuntal music can be found in the literature. [3]. Methods adapted from artificial intelligence have been applied, including genetic algorithms [4], neural networks and connectionism [5], machine learning [6] [7], probabilistic logic [8], and neighborhood search [9]. Stochastic methods naturally play a part in counterpoint composition and analysis [10]. Methods have also been devised for evaluating the artistic merit of random choices in algorithmic music composition [11].

While you could try one of these more challenging approaches, for this assignment it suffices to do a straightforward translation of the contrapuntal rules, using random generation of notes and intervals that abide by the constraints.

Be sure that you document your program with the rules of contrapuntal structure that you have applied for your implementation.

Expected Output of your Program

Your algorithm can provide a variety of output:

- a piece of music that is played as the program runs
- a digital audio file that can be played after the program finishes executing
- a MIDI file that can be input into a MIDI sequencer, played, and transcribed to a musical score

In the implementation that we offer as a possible solution, we have used MATLAB and a MIDI toolbox [12], and we offer all three types of output listed above. The notes in our composition are generated in MATLAB can be output audibly in one of three forms: single-frequency pitch, a square wave, a sawtooth wave, or a triangle (the last three having more interesting timbres than single-frequencies). The MIDI toolbox has a writemidi function that allows us to output our notes in MIDI format as well. The MIDI file allows composition to be played with whatever instrument the user chooses.

References


