Harmonization of Musical Progressions with Genetic Algorithms

Andrew Horner and Lydia Ayers
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, HONG KONG
telephone: (852) 2358-6998, (852) 2355-0558
e-mail: horner@cs.ust.hk, layers@cs.ust.hk

ABSTRACT: This paper introduces a genetic algorithm method for harmonizing complex musical progressions, including those with inverted chords, seventh chords, secondary dominants, chromatic chords and even modulations. We also extend our technique to harmonizing inner voices given both the chord progression and the melody.

Harmonization of musical progressions is a standard music theory problem with fairly strict constraints. We can group these constraints into two classes: those that define how to voice individual chords and those that define how the voices should move from one chord to another.

Individual chord constraints include: only chord tones should appear in the voices; the bass note must agree with the chord's inversion; the voices must lie within their respective ranges; the voices should use standard doublings; the spacing between the upper three voices should not exceed an octave; and the voices should not overlap one another. Voice leading constraints ordinarily prohibit: consecutive parallel octaves, fifths and fourths; direct octaves and fifths (approaching an octave or fifth through similar motion by leaps in both voices); implied voice crossings (an alto note higher than the previous soprano note); unresolved tendency tones, augmented intervals the soprano staying on the same pitch for more than two consecutive notes; large skips; consecutive skips in the same direction; incorrect cadential 2 resolution; incorrect seventh chord resolution; incorrect augmented sixth resolution; and incorrect secondary dominant resolution.

We used these constraints to devise an efficient computer solution to the harmonization problem. Satisfying the voice leading constraints is difficult. If we build a solution one chord at a time from the beginning or end of the progression, we may come to a chord that we cannot approach without changing a chord that the computer has already created.

Our solution relies on breaking the harmonization problem into the subproblems of finding all the possible correct individual chords, and then finding a sequence of these chords that satisfies all the voice leading constraints. We solve the first subproblem by enumeration and the second through genetic algorithm search (Goldberg 1989).

Finding the set of correct individual chords is relatively easy, since there are only a few constraints and no context dependencies. We enumerate all the possible voicings for each chord and keep those that satisfy the individual chord constraints. Since the voice ranges affect which chord voicings will work in each key, we need to find a separate set of voicings for each of the progression's key centers.

After finding all the correct individual chords, we use a genetic algorithm (GA), an optimization procedure based on natural selection, to solve the chord sequencing problem. The GA searches through the set of correct chord voicings until it finds a sequence of voicings that satisfies all the voice leading constraints. The genetic algorithm uses a fitness function to guide its search through the space of possible chord sequences. A fitness function tells the GA how good a particular candidate sequence is. For the chord sequencing problem, the fitness function simply evaluates how many of the the voice leading constraints were violated. For example, if we find two constraints violated, such as consecutive fifths in the bass and
toner starting on the first chord, and direct fills between the same two voices going into the final chord, the GA's fitness function would return a score of -2. The goal is to find a sequence with a fitness of -0.

The following figure shows the GA can handle even complex extended chord sequences, such as frequent modulations, secondary dominants, and chromatic chords. For example, the final chord of the progression is a borrowed chord from e minor.

A common variant of the harmonization problem is 4-part harmonization given both the chord progression and the soprano melody line. The GA finds the appropriate bass line for the chord progression and then the inner two voices. We created a modified version of our program that allows the user to specify the chord progression and the melody line. The GA procedure then works as before, optimizing the chord sequence, but it only reports the existence of soprano-bass voice leading problems without fixing them. We ran the GA on a melody and chord progression including borrowed chords, chromatic chords and modulations to relatively dissonant keys. This is a tough example to challenge the GA with many constraints. The following figure shows the GA's solution to the problem, having filled out the inner voices:

The GA method gives music theory teachers a way to automatically test new progressions to ensure a solution exists. The method also gives arrangers and composers working in the common-practice style a tool for computer harmonization. Those working in extended common-practice style can easily add their own constraints, or modify and delete existing ones (e.g., clearing unessential tritonic tones). Finally, teachers can also use the GA fitness function directly to grade students' solutions. We have encoded a flag that causes the program to report any violated constraints and where they occurred.

References