Analysis of Jazz Chords as Optimization

Here's a formulation of the analysis of jazz chord sequences as an optimization problem[1]. We are given a sequence of n chords $\langle C_1, C_2, ..., C_n \rangle$. From each chord in this sequence, we can compute the set of scales that can be played over it[2]. Let there be m_i such scales for chord C_i, for $1 \leq i \leq n$, and let them be denoted by $\langle S_{i1}, S_{i2}, ..., S_{im_i} \rangle$.

A *solution* to the analysis problem selects one scale for each chord in the sequence, and is represented by $\langle a_1, a_2, ..., a_n \rangle$, where $1 \leq a_i \leq m_i$ for $1 \leq i \leq n$. Its *cost* is defined to be $\sum_{i=1}^{n-1} \Delta(S_{ia_i}, S_{i+1a_{i+1}})$. The function $\Delta(S, T)$ takes two scales S and T and returns the number of notes in S but not in T.

Our objective is to find a solution that minimizes the cost.

To correctly identify II-V's, we can generalize each term in the cost to $\Delta(S_{ia_i}, S_{i+1a_{i+1}}, C_i, C_{i+1})$ and subtract a "bonus" score when the scales S_{ia_i} and $S_{i+1a_{i+1}}$ are equal and C_i and C_{i+1} are the II and V chords, respectively, of the root of that common scale. Favoring the choice of certain scales (like playing the Lydian mode over isolated major chords) can also be handled in this way.

A dynamic programming solution to this minimization problem is simple. Let c_{ip} be the minimum cost that can be attained for the first i chords given that scale S_{ip} is chosen for chord C_i. Then, $c_{i+1q} = \min_{1 \leq p \leq m_i} \{c_{ip} + \Delta(S_{ip}, S_{i+1q})\}$. Initially, we set $c_{1p} = 0$ for $1 \leq p \leq m_1$. The table c_{iq} can then be filled in increasing values of i. An additional table that keeps track of the p chosen when the minimum is taken in the recurrence equation can be used to recover the minimal solution at the end of the computation.

The algorithm runs in time $m_1 m_2 + m_2 m_3 + ... + m_{n-1} m_n$. In other words, if there are always m or fewer choices of scales for each chord, the algorithm will run in $O(m^2 n)$ time.

Alternatively this problem can be viewed as a (single-source) shortest path problem in an acyclic graph. This is, as they say in the business, "left as an exercise".

[2] E.g., an implementation may contain a list of scales; among these, those that contain all the notes of a certain chord are considered playable over that chord.