

## 多数の実例との照合による旋律に対するコード進行付与手法

情報科学科 前田 高志

指導教員：山本 晋一郎

## 1 背景と目的

コードとは音程の異なる複数の音が同時に鳴るときの合成した音のことで、時間の変化に伴いコードが変化することをコード進行という。コードやコード進行は楽曲を大きく色付ける。しかしコードを感覚で自由に扱うには小さい頃からピアノを習熟する等により音感に長けている必要があり、またコードを知識で扱うにも音楽理論を学ぶ必要がある。一方で旋律は和音を使わずに成立するので、直感でも作曲することができる。

旋律にコードを付与する既存のシステム [1][2] は、コード付与にダイアトニックコードのみを使用していたり、旋律の臨時記号に対応していない等の初歩的なコード付けとなっている。また一つの旋律に対して一つのコード進行のみを出力する。一つの旋律に対してあらゆるコード進行が考えられるので、出力されるコード進行が1種類では解答としての幅が狭い。

本研究は旋律を作曲できるがその旋律に適した和音を構成することができない人のために、様々なコードの使用と旋律の臨時記号に対応したコード付けを加味した上で旋律に適したコード進行を複数提示することによる作曲支援が目的である。

## 2 旋律にコード進行を付与する手法の提案

コード進行の実例と旋律を照合し、どの程度各コード進行がその旋律に適しているかを算出する。ただし旋律とコード進行は8小節程度であるとする。それは旋律の終止は8小節程度で組み込まれるからである。旋律とコード進行の照合方法は以下の通りである。

1. 小節内でそれぞれのコードに対応している旋律の音の高さがコードの構成音に含まれている割合を計算し、その数値を小節内の適合率とする。ここでの割合は1小節に対する音の長さである。旋律が休符の場合は無条件で旋律の音がコードの構成音に含まれているとする。
2. 各小節内の適合率を算出した後、各小節の値を乗算した結果を旋律に対してコード進行がいかに対応しているかを示す照合結果とする。

ただし1.において最後の小節に限り、旋律の音に対して不適切な終止形となるコードは、その旋律の音をコードの構成音に含まれていないものとする。

旋律が構成音に含まれていることは旋律とコードが協和することを意味する。またコード進行に実例を使用することで、コード間の移り変わりが適切なコード進行を出力することができる。一小節でも旋律とコードが不協和な小節があればそのコード進行を旋律に対して不適切な進行とするために、小節内の適合率に対して相加重平均ではなく乗算をする。またあらかじめ旋律の経過音と刺繍音と考えられる非和音をあらかじめ和音に変換する。これを行うことによりコードに使うべき音が明確になる。

## 3 実装

提案した手法を実証するためにシステムを構築した。概要を図1に示す。

はじめに旋律を入力する。入力ファイルは旋律を記述した MusicXML 形式を扱う。旋律とコードを照合し、結果を出力するまでの処理を以下に順に記述する。

## 3.1 非和声音の処理

旋律の音で非和声音と考えられる音をコードの評価に含まないように次の2つの処理を行う。

**経過音** ある和音構成音から別の和音構成音へ順次進行でつなぐときに、間に入る非和声音を経過音という [3]。この概念に基づいて、3つの音が短2度もしくは調2度の間隔で連続して音高が高くなるとは低くなる時は、2つ目の音の高さを1つ目の音の高さに、5つの音が連続するときは2つ目の音の高さを1つ目の音の高さに、4つ目の音の高さを3つ目の音の高さに変換する。

**刺繍音** ある和音構成音から、非和声音に順次進行して動き、元の音に戻るとき、その非和声音のことを刺繍音という [1]。この概念に基づいて、ある2つの同じ音程の間にある音符の音程が、2つの音程と短2度か長2度高いまたは低い場合、間にある音符の高さを一つ前の音符の高さに変換する。

## 3.2 度数表記変換

コード進行と旋律をそれぞれ音名表記から度数表記に変換する。これを行うことにより、調に依存しない照合を可能にする。

## 3.3 照合

2節の手順に従って旋律とコード進行を照合する。

## 3.4 終止の考慮

入力旋律の最後の小節は曲の終わりや区切りと判断でき、音楽理論でそれは終止と呼ばれる [3]。終止の旋律から不適切な終止を導かないように以下の処理を加える。

- VIm のコードは I の旋律の音を構成音と評価しない。

- VIm のコードは III の旋律の音を構成音と評価しない。
- I のコードは V の旋律の音を構成音と評価しない。

## 3.5 照合結果の出力

照合結果を確認するためのテキストファイルを作成する。照合結果、調、曲名、アーティスト名、楽曲形式の場所、小節数、コード進行のナンバーを一つのコード進行の情報とし、照合結果が高い順にコード進行の情報をテキストファイルに記述する。

## 3.6 コード進行の MIDI ファイル生成

コード進行の音を確認するために MIDI ファイルを作成する。3.5 で与えられたコード進行のナンバーと欲しい調を入力し、その調性に合わせられたコード進行の MIDI ファイルを Haskore ライブラリ [4] を使用して出力する [5]。

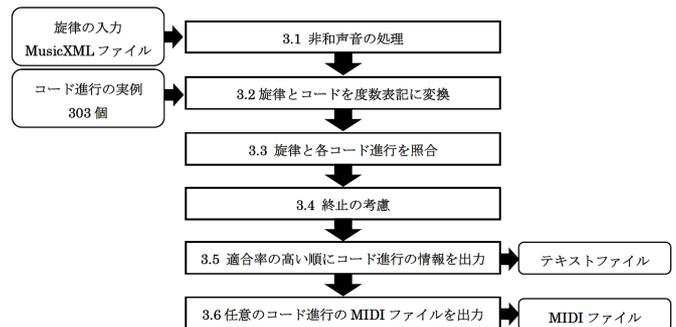


図1 コード進行付与システムの概要

## 4 評価

実例のコード進行に使われている旋律を入力し、そのコード進行が実例全体の何番目の順位で出力されるかを評価する。今回評価には303の実例となるコード進行と15の旋律を用意し、それぞれの旋律に実際に付けられたコード進行の順位の平均をとる。実際に付けられたコード進行は303の実例の中に含まれているものとする。入力としてクラシック音楽の旋律を4つと、1978年から2008年までの日本のポピュラー音楽の11の旋律を使用した。旋律は臨時記号が含まれているものも存在し、全て8小節で4分の4拍子のものである。提案手法の検証のための、他の3つの手法と比較する。順位は小数第4位を四捨五入する。評価結果を表1に示す。

表1 評価結果

手法	平均順位	最低順位	順位が1位の数
提案手法	1.7	9	11
小節間の適合率の相加重平均	2.0	9	10
非和声音を考慮しない	2.2	10	11
終止を考慮しない	2.3	13	9

結果から、小節間の適合率の相加重平均ではなく乗算を行い、非和声音と終止を考慮した提案する手法が最も平均順位が良く、およそ1.7位に出力されることがわかった。15の旋律のうち11の旋律は実際に付けられたコード進行が実例のうち最も適したコード進行と出力した。

## 5 まとめと今後の課題

和声理論によるルールに基づいたコード付けではなく、実例との照合によるコード付けを行うことで、様々なコードの使用や臨時記号の対応という高度なコード付けにも対応させた。また実例のあるだけの結果を出力することにより、一つの旋律に対して複数のコード進行の結果として出力することができ、ユーザに幅の広い解答を用意することができた。

今後の課題として現在は8小節のコード進行のみを用意したので、8小節以外の旋律にも対応できるようにコード進行を用意するべきである。旋律に実際に使われたコード進行以外の照合結果の高いコード進行も評価する必要がある。

## 参考文献

- [1] 三浦雅展 他: “ポップス系の旋律に対する和声付与システム:AMOR,” 情処論, Vol. 46, No. 5, pp. 1176-1187, 2005.
- [2] Kooops, H.V., J.P.Magalhaes, and W.B. de Haas, “A Functional Approach to Automatic Melody Harmonisation”, In Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & design (FARM '13), 2013.
- [3] 山縣 茂太郎 『新訂 音楽通論』 音楽之友社, 1958.
- [4] Haskore, <http://www.haskell.org/haskellwiki/Haskore>.
- [5] Jeremy Gibbons and Oege de Moor, 『関数プログラミングの楽しみ』, 山下伸夫訳, オーム社, 2010.