コード進行推薦による作曲支援システムの開発

ハーモニーサーチを用いたコード推薦について

1. はじめに

本稿では, コード進行推薦アプリケーションの新しい推薦方法として採択したハーモニーサーチアルゴリズム(Harmony Search Algorithm, HS)を紹介し, コード進行推薦への応用について述べる.

2.　Harmony Search Algorithm

ハーモニーサーチ(Harmony Search Algorithm, HS)とは, 音楽家の即興過程を模倣した最適解探索アルゴリズムである. HSの基本的な流れは既存の解から新しい解を作り, 更新していくという, 遺伝的アルゴリズム(Genetic Algorithm, GA)と似ている.しかし, GAが2つの親だけを考慮対象とし新しい解を生成することに対して, HSは, ハーモニーメモリー(HM)に入っている既存のすべての解を考慮した後. 新しい解を生成するので, GAより色んな可能性に触れることができ, より柔軟性の高い処理ができると考えられる. HSは大きく5つのStepで構成される.

**Step1. 初期値設定**

　・決定変数の設定

・各決定変数の値の範囲設定

・Harmony Memory Size(HMS)設定

・Harmony Memory Considering Rate(HMCR)設定

・Pitch Adjusting Rate (PAR)設定

・Termination Criterion設定

・目的関数(objective function)設定

**Step2. Harmony Memory(HM)初期化**

・HMSを基にランダム的にHMを初期化

・目的関数の値順にsort

テーブル

自動的に生成された説明

**Step3. 新しいハーモニーの生成**

・HMCR, PARとRandomizationを利用して生成

・生成方法は以下の2つ

1. 既存のHMからハーモニーを選択する

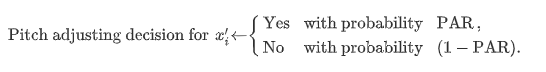
テキスト, 手紙

自動的に生成された説明

例えば, HMCRを0.85と設定した場合、85％の確率で, 新しいハーモニーのi番目はHMのi列の決定変数の値から選択される. 残り15%の確率で, HM内全ての決定変数の値から選択される.

　2） ハーモニーのピッチを調整する

獲得したハーモニーに対して, ピッチを調整する必要があるかどうかを判断する. この操作では, 以下のようにピッチ調整に関わるPARパラメーターを使う.



グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

**Step4. HMのアップデート**

　・新たに生成されたハーモニーがHM内のハーモニーより優秀なら, そのハーモニーをHMに入れ, HM内の最も悪いハーモニーは削除する

・目的関数の値順にsort

**Step5. Termination Criterionをチェック**

・終了条件を満たしたら終了

・そうじゃない場合はStep3へ戻る

以上のStepを繰り返えしながら最適解を探すことになる.

3.　コード進行推薦への応用

本研究では, 先に説明したHSを用いてコード進行推薦への応用を考えている. HSは元々最適化探索のためのアルゴリズムであるが, 本研究では, 最適の解一つだけを残すという「結果」ではなく, 最適の解を求めていく過程で得られる「最適解に近接した候補」を提示することを目標としている. コード進行推薦という観点から説明すると, 現在のコード進行と最も類似度が高いコード一つだけを答えとして提示するのではなく, 適当に高い類似度を持つ多数のコードを提示することである.それらの候補の中から選んでいくことで, コード進行を作ることになる.

さて, HSを用いるにあたって必要な目的関数を以下のように定義している.

*harmony(current) = {*

*harmony(new) = {*

ここで, *harmony(current)* は現在のコード進行, *harmony(new)* は新たに生成されたコード進行を指す. やなどはそのコード進行の要素(つまり, コード)を表す. すなわち, 目的関数の値は現在のコード進行と新しいコード進行間の距離で求めあれる. 距離については, 五度圏という音楽理論に基づいて計算する.

*グラフ が含まれている画像

自動的に生成された説明*

図 1：五度圏

　また, 五度圏による距離計算においては, 以下の二つの点を補足する.

　1）距離が短い方を採択する

例)CからDまでの距離を求める方法として

a）C→G→D：2

b）C→Am→Em→Bm→D：4

　　のような方法が考えられるが最も短いaをCからDまでの距離として採択する.

　2）正反対にはコスト１で行ける

　　正反対側とは, 代理コードという特別な関係があるので, コスト1で行けるように設定.

例) FからBまでの距離は6ではなく1.

　上記のルールをまとめると, 以下のような十二

角形のグラフで距離を求める問題となる.

グラフ, レーダー チャート

自動的に生成された説明

図 2：コード間距離グラフ

　図2のグラフから最短距離を測ることで目的関数の値が求めあれる.

4.　まとめ・今後の課題

以上で, コード進行推薦アプリケーションの新しい推薦方法として採択したハーモニーサーチアルゴリズム(Harmony Search Algorithm, HS)を紹介し, コード進行推薦への応用について述べた. 今後の課題としては, 目的関数の再検討と, 既存の推薦システムとの結合が考えられる. 目的関数の再検討については, 現在の目的関数は単にユークリッド距離の表現法であって, 五度圏による最短距離の計算という実際行っていることとは乖離がある. したがって目的関数の再検討が必要だと考えられる. また, 既存の推薦システムの各機能を考慮しうまく結合することも課題として考えられる.

参考文献

1. Kang Seok Lee, Zong Woo Geem : A new structural optimization method based on the harmony search algorithm, Computers and Structures 82(2004) 781-798.