「転調可能な純正律」を目指して～プログラミングを用いた音律最適化と応用～

城ノ内中等教育学校

動機

かつて周波数を 2 倍した音をオクターヴと定め、3/2 倍、5/4 倍した音を組み合わせて**純正律**を作った。
※C(ド)と高い C(ド)のような音名が同じ関係。

音の比が簡単な整数倍で表せきれいに響く一方で、転調できない。そこで現代ではオクターヴを 12 等分し割り当てる**平均律**が使われている。
※全体として使用する音を上下に平行移動させること。
※比を等分するため 2 のn/12 乗。

この方法はすべての調に転調可能な一方、**響きの「美しさ」が妥協されている。**楽曲内で使用する調に限定し、評価することで転調可能かつ**より最適な音律が作れないか**と思いこの題を定めた。
※オクターヴの間の音をどのような比で割り当てるか

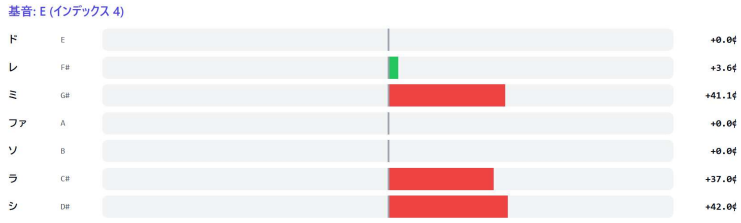

図 1　転調時の純正律のズレ(cent)
※cent(¢):1 オクターヴを 1200 等分したものを 1 とした値

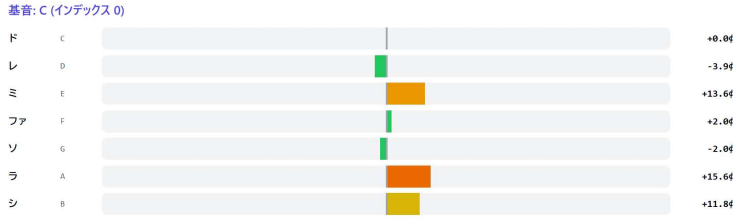

図 2　純正律と平均律のズレ(cent)

実装

言語はライブラリの充実した python を用いる。

任意の使用する調の集合を与えると、**焼きなまし**で初期会を生成し、
※ランダムに少しずつずらした比を生成し、スコアが改善したら採用、しなくても確率で採用する、局所解に陥りづらい探索方法

scypy.optimizeで最適化する。
※規則に則って少しずつずらした比を生成し、スコアが改善したときのみ採用する単峰性がある確率変数に対して有用な探索方法を自動でしてくれるライブラリ

極端な解を避けるため評価関数を以下のように定義する。

```
R = np.array([1,16/15,9/8,6/5,5/4,4/3,45/32,3/2,8/5,5/3,9/5,15/8,2])
nD = {0,2,4,5,7,9,11,12}
target_logs = np.log2(R)
w = 0.0001
def score_for_A(A, Bs):
    total = 0.0
    for Bi in Bs:
        b = A[Bi]
        e = []
        for j in range(13):
            p = Bi+j      比なのでそのままでは引き算に意味がないため、log を取った状態で純正律との差を求める
            d = A[p%12] + p//12 - b - target_logs[j]
            e.append(d*d if j in nD else d*d*w)
        total += np.mean(e)  差の 2 乗を追加することで、突出したズレを抑える
    return total / len(Bs)   半音でも微細な重みで評価に加え、孤立した比を抑制する
```

結果

平均律の初期スコア:921.38

半音上下する転調がある楽曲において**スコア 0.60 改善率 99.9%**

全音上下する転調がある楽曲において**スコア 467.66 改善率 49.2%**

どの調においても大きな優位性が見られた(図 2)。

音差	±1	±2	±3	±4	±5	±6
スコア	**0.6**	**467.66**	363.06	**0.35**	380.18	261.28
改善度	**99.93%**	**49.24%**	60.60%	**99.96%**	58.74%	71.64%

図 2　音の差(半音)のごとのスコアと改善度

応用

今回あえて静的な音律を作ったのは**汎用的な応用**が目的である。

1. **.mid ファイルから調を解析、チューニング済音声ファイルの作成**
※音の高さ、タイミング等を示した楽譜や作曲でよく使われるファイル
python で.mid を読み取り、**自動で反映した音声を生成する**プログラムを作成。

2. **MIDI 鍵盤の入力に対応**
※パソコンに接続し、音程や強さなどの演奏情報をデータとして送るピアノ型の入力装置
まず調を指定し音律の作成、**キーボードの入力をリアルタイムで変換し再生する**プログラムを作成。

3. **DTM への応用**
※パソコンを用いて作曲すること
自動でチューニングファイル(.scl)を生成し、DAW ソフトで生成した音律を使用可能にする。例えば Vital だと.scl を読み取ることができるため、作成した音律を楽器に反映させることが可能となる。他の DAW ソフトへの連携、新規インストゥルメントの生成も可能であるため高い応用力をもつ。
※パソコンでの作曲に用いられるソフトウェア
※DAW ソフト上で用いることのできる楽器

展望

1. **GUI の作成**
現在コンソールアプリケーションであり使える人が限られる
※簡素な画面に文字を入力すると、文字を返すシンプルなアプリケーション
→視覚的にわかりやすいアプリを

2. **オープンソース化**
プログラムを公開しだれでも使える汎用的なものにする

参考文献:
Pythagorean Tuning and Medieval Polyphony(https://www.medieval.org/emfaq/harmony/pyth.html)
Just Intonation Explained - By Kyle Gann(https://www.kylegann.com/tuning.html)
A MATHEMATICAL MODEL FOR OPTIMAL TUNING SYSTEMS
(https://eamusic.dartmouth.edu/~larry/published_articles/owt_pnm.pdf)
Playing Music in Just Intonation – A Dynamically Adapting Tuning Scheme
(https://arxiv.org/pdf/1706.04338)